2001

Adaptive filters for acoustic echo cancellation and noise reduction in a car.

Wayne Hui Chung. Chiang
University of Windsor

Adaptive Filters for Acoustic Echo Cancellation and Noise Reduction in a Car

by

Wayne Hui-Chung Chiang

A Thesis

Submitted to Faculty of Graduate Studies & Research

Through the Faculty of Engineering – Electrical & Computer Engineering

In Partial Fulfillment of the Requirements for

The Degree of Master of Applied Science at the

University of Windsor

Windsor, Ontario, Canada

December, 2000

0-612-62201-0

928765

Abstract

In recent years, hands-free telephony systems have experienced growing and great interest for convenience and safety reasons. A non-interference speech signal is an ideal for the hand-free telephony communication system. Because prior to reception, the signal is likely to be unknown, a conventional and simple fixed filter is not useful in this application. Therefore, for reducing acoustic echoes and noise, the cancellers are designed with adaptive transversal FIR digital filters, and based on variants of the least mean square (LMS), recursive least square (RLS) and normalized least mean square (NLMS) algorithms. The ability of updating tap-weights in adaptive filters is suitable for cancelling non-stationary acoustic echo and noise. The experiments tested these three adaptive algorithms with different sets of data, and shown is the comparison of the three algorithms.

Dedicated with love

to my mom and my family,

Acknowledgment

I would like to express my sincere gratitude to my thesis advisor Dr. Kwan for his invaluable guidance, discussions and comments, and constant encouragement throughout the process of this thesis. In particular, I would like to acknowledge him for suggesting adaptive echo and noise cancellation as the topic of my thesis and to thank him for going through the first and final drafts of the thesis, and for making various suggestions to improve the presentation of the thesis. I would also like to thank my internal reader, Prof. P. H. Alexander, and my external reader, Dr. Y. H. Tsin for their comments in enhancing the thesis presentation.

I would also like to thank Ms. Shirley Ouellette and Mr. Alan Johns' for their assistance over the years. I would like to sincerely thank all my friends, especially, Tracy Li, Walter Jin, Elaine Lin, Matt Rau, and Dr. Kao's family for their encouragement and friendship during my master's study.

Last but not least, my thanks must be given to my family for their support. My special thanks are given to my mother for her financial support and unconditional love over the years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

LMS	Least Mean Squared Error
RLS	Recursive Least Squared Error
NLMS	Normalized Least Mean Squared Error
FIR	Finite Impulse Response
AEC	Acoustic Echo cancellation
NR	Noise Reduction
RM	Receive Mode
TM	Transmit Mode
DT	Double-Talk Mode

Chapter 1

1 Introduction

1.1 Combined acoustic echo cancellation and noise reduction

In this high–tech age, digital voice systems are being used in a variety of environments, and their performance must be maintained at a level near that of noise-free speech. This is the task that many researchers are investigating. This thesis will concentrate on the area of the car environment.

In terms of safety, comfort and convenience, a hands-free telephone communication device can be a very desirable feature in a car. Both echoes and additive noise cause the quality of speech recognition to be reduced and result in inconvenience to the far-end listener.

1.2 A Definition of Hands-free

Hands-free phones provide what we are already used to with conventional phones, without having to hold a handset. Thus instead of bringing ourselves to a handset, we bring the elements of a handset to ourselves, namely that of a microphone and a loudspeaker. Thus with the removal of the constraint of a handset, one is now able freely to circulate within one's environment.

Apart from the obvious uses within cars, offices, homes, and for teleconferencing, hands-free technology can literally be a lifesaver. Those home-bound people with disabilities stand to benefit both psychologically and medically by being able to communicate easily with the outside world. The ability of communication within the home we all take for granted is a very precious tool to the disabled, especially those living on their own; studies have shown that short conversations with loved ones can dramatically change a person's mood. Additionally, hands-free technology would also be of great importance in the event of a medical emergency, when help needs to be summoned as quickly and as easily as possible. Also, remote medical surgery being pioneered in the USA would also benefit, as the remote surgeon performing the operation could easily communicate with the medical staff on the patient's side. Thus not only does hands-free technology offer convenience to the millions, it also offers life to the thousands.

1.3 Acoustic echo cancellation

Hands-free systems are mainly disturbed by the acoustic echo signal that originates from the sound propagation between the loudspeaker and the microphone in an audio terminal. Current solutions for removing the echo are based on the real-time identification of the acoustic impulse response and uses adaptive filtering. Unfortunately, there are three difficulties with this technique: first, the acoustic echo path has a relatively long duration that translates into a requirement for huge amounts of computations and memory; second, the echo path can vary very quickly in time; and finally, the output disturbance as well as the input signal are non-stationary (speech signals).

In a telephone communication, the echo can be categorized into two types of echo, which are line echo and acoustic echo. Line echo is caused by a long distance signal transmission. This problem is usually solved by the telephone company. The acoustic echo is due to the coupling of the reflection of the loudspeaker's sound from the car ceiling, windows, floor and many other objects back to the microphone. This phenomenon is not only present in a mobile situation, but also in some other areas such as audio/video conferencing and in the case of a hearing-aid.

This paper's approach on the acoustic echo cancellation uses an adaptive filter. The adaptive filtering is trying to retain a good estimate of the echo. And the replica echo is then subtracted from the near-end signal. A good algorithm should have a good result and fast computing time.

1.4 Noise Reduction

One of the main problems with hands-free operation in a car is related to the high background noise level. The additive car noise in a car could be from the engine, road, wind, bumps, noise when passing a car running in the opposite direction, etc.
An effective method should be applied on the noise reduction in order to obtain a good telecommunication or telephone devices voice recognition.
Many methods have been employed on noise reduction, such as spectral subtraction and the use of microphone arrays. Spectral subtraction has well-known disadvantages such as limited performance at low SNR values and artificial sounding residual noise.
Microphone arrays could have a good result, but due to the number of microphones used in the array, this is not practical.

Adaptive signal processing evolved from techniques developed to enable the adaptive control of time- varying systems. It has gained a lot of popularity due to the advances in digital technology that have increased the computing capacities and broadened the scope of digital signal processing. The key difference between classical signal processing techniques and adaptive signal processing method is that in the latter we deal with time varying digital systems. When adaptive filters are used to process non-stationary signals, whose statistical properties vary in time, the required amount of prior information is often less than that required for processing via fixed digital filters. An adaptive filter is a good tool in application to non-stationary speech signal processes.

1.5 About the thesis

1.5.1 The objectives

This thesis looks at a car telecommunication problem involving echoes and noise. Appropriate adaptive filters (FIR), algorithms (RLS, LMS and NLMS) are applied to tackle the problem successfully. The work presented is trying to apply the adaptive filters to substitute for the conventional spectral subtraction and also deal with both acoustic echoes and additive background noise at the same time.

1.5.2 The organization

In Chapter 2, we discuss the concept of an adaptive interference cancellation filter. An introduction is given on the adaptive filter. The input speech and reference signals are also discussed along with the filter.

In Chapter 3, we give detail of the LMS and RLS algorithms.

In Chapter 4, we give detail of the Normalized LMS algorithm, a better version of LMS.

In Chapter 5, Systems Design, four modes are discussed. It is also applied adaptive filter but with four separate steps which are IDLE mode, Receive mode, Transmit Mode, and Double-talk mode.

In Chapter 6, the experiments are designed for testing a pre-recorded database. The comparison and analysis result of using LMS, RLS and NLMS are presented.

In Chapter 7, the conclusions and some future direction on the subject are discussed.

Chapter 2

2 Adaptive Filter Structure

2.1 The Filtering Problem

The term filter is often used to describe a device in the form of a piece of physical hardware or software that is applied to a set of noisy data in order to extract information about a prescribed quantity of interest. The noise may arise from a variety of sources. For example, the data may have been derived by means of noisy sensors or may represent a useful signal component that has been corrupted by transmission through a communication channel. In any event, we may use a filter to perform three basic information-processing tasks [2]:

1. Filtering, which means the extraction of information about a quantity of interest at time t by using data measured up to and including time t.
2. Smoothing, which differs from filtering in that information about the quantity of interest need not be available at time t, and data measured later than time t can be used in obtaining this information. This means that in the case of smoothing there is a delay in producing the result of interest. Since in the smoothing process we are able to use data obtained not only up to time t but also data obtained after time t, we would expect smoothing to be more accurate in some sense than filtering.
3. Prediction, which is the forecasting side of information processing. The aim here is to derive information about what the quantity of interest will be like at some time $t + \tau$ in the future, for some $\tau > 0$, by using data measured up to and including time t.

We may classify filters into linear and nonlinear. A filter is said to be linear if the filtered, smoothed, or predicted quantity at the output of the device is a linear function of the observations applied to the filter input. Otherwise, the filter is nonlinear.

2.2 Adaptive Filters

The design of a Wiener filter requires a priori information about the statistics of the data to be processed. The filter is optimum only when the statistical characteristics of the input data match the a priori information on which the design of the filter is based. When this information is not known completely, however , it may not be possible to design the Wiener filter or else the design may no longer be optimum.[7] A straightforward approach that we may use in such situations is the "estimates and plug" procedure. This is a two-stage process whereby the filter first "estimates" the statistical parameters of the relevant signals and then "plugs" the results so obtained into a nonrecursive formula for computing the filter parameters. For real-time operation, this procedure has the disadvantage of requiring excessively elaborate and costly hardware. A more efficient method is to use an adaptive filter. By such a device we mean one that is self-designing in that the adaptive filter relies for its operation on a recursive algorithm, which makes it possible for the filter to perform satisfactorily in an environment where complete knowledge of the relevant signal characteristics is not available. The algorithm starts from some predetermined set of initial conditions, representing, whatever we know about the environment. Yet, in a stationary environment, we find that after successive iterations of the algorithm it converges to the optimum Wiener solution in some statistical sense. In a nonstationary environment, the algorithm offers a tracking capability, in that it can track time variations in the statistics of the input data, provided that the variations are sufficiently slow.

As a direct consequence of the application of a recursive algorithm whereby the parameters of an adaptive filter are updated from one iteration to the next, the parameters become data dependent. This, therefore, means that an adaptive filter is in reality a nonlinear device, in the sense that it does not obey the principle of superposition. Notwithstanding this property, adaptive filters are commonly classified as linear or nonlinear. An adaptive filter is said to be linear if the estimate of a quantity of interest is computed adaptively (at the output of the filter) as a linear combination of the available set of observation applied to the filter input. Otherwise, the adaptive filter is said to be nonlinear.

A wide variety of recursive algorithms have been developed in the literature for the operation of linear adaptive filters. In the final analysis, the choice of one algorithm over another is determined by one or more of the following factors [7]:

- *Rate of convergence.* This is defined as the number of iterations required for the algorithm, in response to stationary inputs, to converge "close enough" to the optimum Wiener solution in the mean-square sense. A fast rate of convergence allows the algorithm to adapt rapidly to a stationary environment of unknown statistics.
- *Misadjustment.* For an algorithm of interest, this parameter provides a quantitative measure of the amount by which the final value of the mean-squared error, averaged over an ensemble of adaptive filters, deviates from the minimum mean-squared error that is produced by the Wiener filter.

- *Tracking.* When an adaptive filtering algorithm operates in a nonstationary environment, the algorithm is required to track statistical variations in the environment. The tracking performance of the algorithm, however, is influenced by two contradictory features: (1) rate of convergence, and (2) steady-state fluctuation due to algorithm noise.
- *Robustness.* For an adaptive filter to be robust, small disturbances (i.e., disturbances with small energy) can only result in small estimation errors. The disturbances may arise from a variety of factors, internal or external to the filter.
- *Computational requirements.* Here the issues of concern include (a) the number of operations (i.e., multiplications, divisions, and additions/subtractions) required to make one complete iteration of the algorithm, (b) the size of memory locations required to store the data and the program, and (c) the investment required to program the algorithm on a computer.
- *Structure.* This refers to the structure of information flow in the algorithm, determining the manner in which it is implemented in hardware form. For example, an algorithm whose structure exhibits high modularity, parallelism, or concurrence is well suited for implementation using very large-scale integration (VLSI).
- *Numerical properties.* When an algorithm is implemented numerically, inaccuracies are produced due to quantization errors. The quantization errors are due to analog-to-digital conversion of the input data and digital

representation of internal calculations. Ordinarily, it is the latter source of quantization errors that poses a serious design problem. In particular, there are two basic issues of concern: numerical stability and numerical accuracy. Numerical stability is an inherent characteristic of an adaptive filtering algorithm. Numerical accuracy, on the other hand, is determined by the number of bits (i.e., binary digits) used in the numerical representation of data samples and filter coefficients. An adaptive filtering algorithm is said to be numerically robust when it is insensitive to variations in the wordlength used in its digital implementation.

Speech signal processing, signal modelling, wiener filtering, and spectrum estimation are all made an important assumption that the signals that were being studied were stationary. Unfortunately, many signals we are dealing with in the real world are not really stationary. In order to deal with the non-stationary or rapid changing signals, the adaptive filter has coming to the picture.

Noise cancelling is an application that is highly taking advantage of the adaptive filter. Filters used for noise cancelling purpose can be fixed or adaptive. The design of fixed filters must be based on prior knowledge of both the signal and the noise. Knowing the characteristics of either or both signal and noise is quite impossible to succeed. But adaptive filters have the ability to adjust their own parameters automatically according the signal and noise, and the filters need little or no prior knowledge of the signal or the noise and still accomplish a very good noise cancelling job. In the adaptive filter process, a reference noise is filtered to produce an output which is a close replica of additive noise which is the cause of corrupted signal. As a result of corrupted signal subtracted by the replica, the noise is eliminated by cancellation.

It seems to be a dangerous process, that of subtracting the noise from a corrupted signal. If done improperly, it could result in increasing in output noise power. [2]

2.3 Adaptive Interference Cancelling

2.3.1 Introduction

An adaptive system is one that is designed primarily for the purpose of adaptive control and adaptive signal processing. Such a system usually has some or all of the following characteristics: [2]

1. They can automatically adapt in the face of changing environments and changing system requirements.

2. They can be trained to perform specific filtering and decision making tasks, i.e., they can be programmed by a training process. Because of this adaptive systems do not

require the elaborate synthesis procedures usually needed for non-adaptive systems. Instead, they tend to be self designing.

3. They can extrapolate a model of behaviour to deal with new situations after having been trained on a finite and often small number of training signals or patterns.

4. To a limited extent they can repair themselves, i.e., adapt around certain kinds of internal defects.

5. They are more complex and difficult to analyze than non-adaptive systems, but they offer the possibility of substantially increased system performance when input signal characteristics are unknown or time varying.

The usual method of estimating a signal corrupted by additive noise is to pass the composite signal through a filter that tends to suppress the noise while leaving the signal relatively unchanged.

2.3.2 Early Work in Adaptive Interference cancelling

Some of earliest work in adaptive interference cancelling was performed by Howells and Applebaum and their colleagues at the General Electric Company between 1957 and 1960. They designed and built a system for antenna sidelobe cancelling using reference input derived from an auxiliary antenna and a simple two-weight adaptive filter.
At the time of this early work, only a handful of people were interested in adaptive systems, and the development if the multiweight adaptive filter was just beginning. In 1959, Widrow and Hoff at Stanford University were devising the least-mean-square (LMS) adaptive algorithm and the pattern recognition scheme known as Adaline (for "adaptive linear threshold logic element").

2.3.3 The Concept of Adaptive Noise Cancellation

The basic concept of adaptive interference cancellation is illustrated in Fig. 2.1. A signal s is transmitted over a channel to a sensor that receives the signal plus an uncorrelated noise, n. The combined signal and noise, corrupted signal $d = s + n$, form the "primary input " to the canceller. A second sensor receives a noise signal x, which is uncorrelated with the signal but correlated in some unknown way with the noise n. The second sensor provides the "reference input" to the canceller. The noise x is filtered to produce an output y that is a close replica of n. This output is subtracted from the primary input $d = s + n$ to produce the system output $e = d - y$.

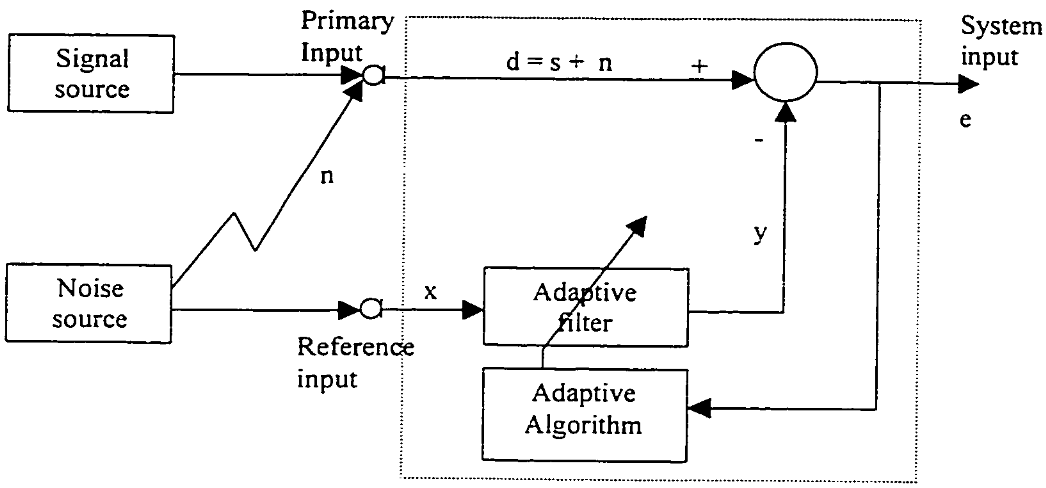

Figure 2.1 Adaptive interference cancelling concept

2.4 The Theory of Finite Impulse Response (FIR)

2.4.1 Issues in Filter Design

The general process of designing a digital filter (for realization either in hardware or software) involves the following four basic steps [2]:

1. Solve the approximation problem to determine filter coefficients that satisfy performance specifications.
2. Choose a specific structure in which the filter will be realized and quantize the resulting filter coefficients to a fixed word length.
3. Quantize the digital filter variables, i.e. the input, output, and intermediate variable word lengths.
4. Verify by simulation that the resulting design meets given performance specifications.

The results of step 4 generally lead to revisions in step 2 and step 3 in order to meet specification.

Although it would be desirable to be able to perform steps 1 to 3 simultaneously, i.e., to be able to solve the approximation problem for arbitrary structures, with arbitrary word lengths, it is not likely that such a design procedure will be available in the foreseeable future. Thus, for the time being we must be content to solve each of these problems independently.

2.4.2 Discussion of FIR Filter

There are many reasons for studying how to design FIR filters. Among the advantage [2] of FIR filters are

1. FIR filters with exactly linear phase can be easily designed. This simplifies the approximation problem, in many cases, when one is only interested in designing a filter that approximates an arbitrary magnitude response. Linear phase filters are important for applications where frequency dispersion due to nonlinear phase is harmful – e.g., speech processing and data transmission.
2. Efficient realization of FIR filters exists as both recursive and non-recursive structures.
3. FIR filters realized nonrecursively, i.e., by direct convolution, are always stable.
4. Roundoff noise, which is inherent in realization with finite precision arithmetic, can easily be made small for non rececursive realizations of FIR filters.

In the next section the properties of FIR filters with linear phase are derived and then a discussion of several design techniques for FIR filters will be in later chapters.

2.4.3 FIR Filter Structure

The M tap FIR filter consists of (M-1) delayers, M multiplyers, each with its correspondent weight, and M-1 adders:

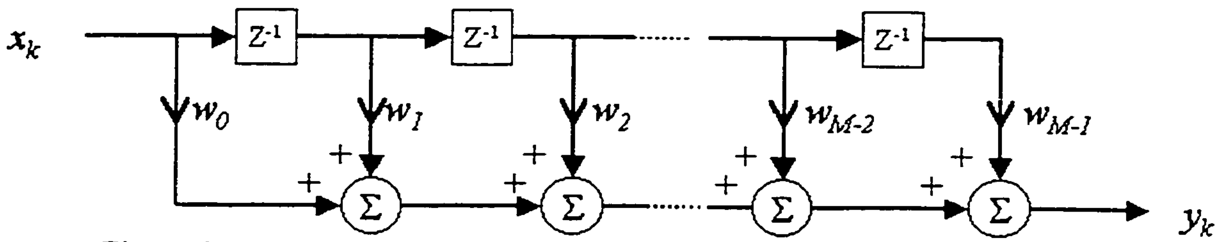

Figure 2.2 FIR Filter Structure

The values of the weights determine the frequency response of the filter. Algebraically, the expression for the output is:

$$y_k = \sum_{j=0}^{M-1} x_{k-j} w_j \tag{2.1}$$

y_k can also be expressed in vectorial notation with X_k and W as vectors:

$$y_k = W^T X_k \tag{2.2}$$

The name FIR stands for Finite Impulse Response. Such is the characteristic of the impulse response of FIR filters, it is finite in duration. Actually it has a duration of M, where M is the number of taps of the filter, and the weights are the values of the M samples of the impulse response.

Chapter 3

3 LMS and RLS Algorithms

At each sampling time, an adaptation algorithm adjusts the filter coefficients to minimise the difference between the filter output and a desired or target signal. The adaptation formula has the general recursive form:

Next Parameter Estimate = Previous Parameter Estimate + Update(error)

In adaptive filtering, decisions have to be made concerning the filter model and the adaptive algorithm. These are [2]:

(a) Filter Type: This can be a finite impulse response (FIR) filter, or an infinite impulse response (IIR) filter. In this project we only consider FIR filters as they have good stability and convergence properties and for this reason are the type most use in practice.

(b) Filter order: Often the correct number of filter taps is unknown. The filter is either set using prior knowledge of the input and the desired signals, or it may be obtained by monitoring the changes in the error signal as a function of the increasing filter order.

(c) Adaptation algorithm: The two most widely used adaptation algorithm are the least mean squared error (LMS) and the recursive least squared error (RLS).

3.1 LMS Adaptive Filter

3.1.1 Introduction

The LMS algorithm is the most popular technique for adaptive filtering applications. Its simplicity and ease of implementation make this algorithm an attractive solution for many practical problems: most commercially available acoustic echo cancellers are based on it.

The LMS algorithm was first introduced to the engineering community in a land mark paper by Widrow and Hoff in 1960. The LMS has found application in many areas such as noise control and adaptive antenna design. It is based on a gradient technique for minimizing the minimum mean-squared error (MMSE). That of coefficient values at the optimal solution is called the MMSE solution. The goal of the adaptive process is to adjust the filter coefficients in such a way that they move from their current position toward the MMSE solution.

The least-mean-square (LMS) algorithm is a linear adaptive filtering algorithm that consists of two basic processes:

1. A filtering process, which involves (a) computing the output of a transversal filter produced by a set of tap inputs, and (b) generating an estimation error by comparing this output to a desired response.
2. An adaptive process, which involves the automatic adjustment of the tap weights of the filter in accordance with the estimation error.

Thus, the combination of these two processes working together constitutes a feedback loop around the LMS algorithm, as illustrated in the block diagram of figure 3.1 (a). First, we have a transversal filter, around which the LMS algorithm is built; this component is responsible for performing the filtering process. Second, we have a mechanism for performing the adaptive control process on the tap weights of the transversal filter, hence the designation "adaptive weight-control mechanism" in figure 3.1 (a).

Details of the transversal filter component are presented in figure 3.1 (b). The tap inputs $x(n), x(n-1), \ldots, x(n-M+1)$ from the elements of the M-by-1 tap-input vector $x(n)$, where M-1 is the number of delay elements; the tap-weight $w_0(n), w_1(n), \ldots, w_{M-1}(n)$ form the elements of the M-by-1 tap-weight vector $w(n)$. The value computed for the tap-weight vector w(n) using the LMS algorithm represents an estimate whose expected value approaches the Wiener solution.

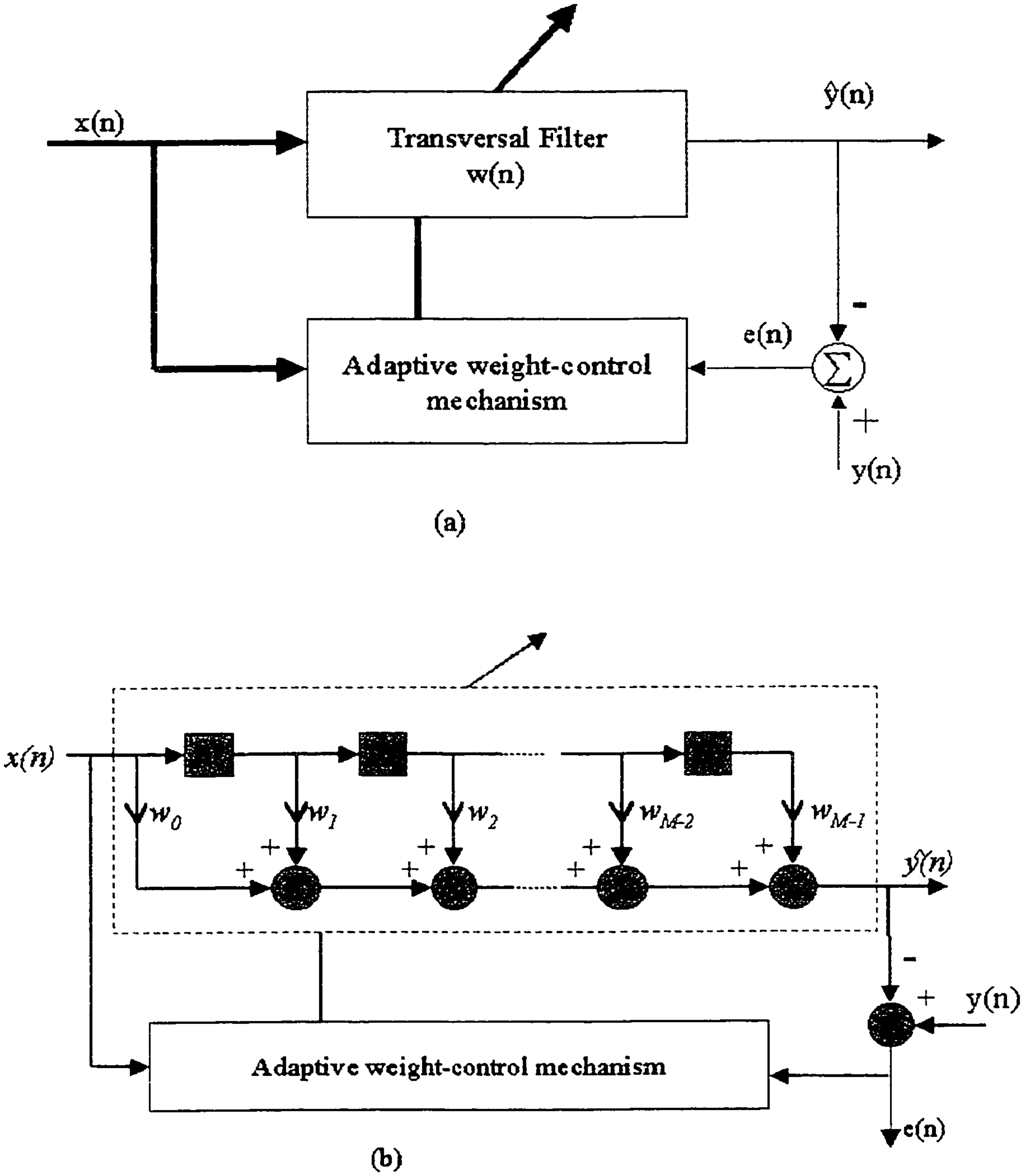

Figure 3.1 (a) Block diagram of adaptive transversal filter. (b) Detailed structure of transversal filter component.

In designing a FIR adaptive filter, the goal is to find the vector $W(n)$ at time step n that minimizes the quadratic function

$$\varepsilon(n) = E\{|e(n)|^2\} \tag{3.1}$$

3.1.2 Derivation of LMS

The filter output can be expressed as

$$\hat{y}(n) = w^T(n)x(n) \tag{3.2}$$

where $\hat{y}(n)$ is an estimate of the desired signal $y(n)$. The filter error signal is defined as

$$\begin{aligned} e(n) &= y(n) - \hat{y}(n) \\ &= y(n) - w^T(n)x(n) \end{aligned} \tag{3.3}$$

The adaptation objective is to minimize the mean squared error defined as

$$\begin{aligned} \varepsilon(n) &= E\{|e(n)|^2\} \\ &= E\{[y(n) - w^T(n)x(n)]^2\} \end{aligned}$$

Thus, in minimizing $\varepsilon(n)$, to find the coefficients that minimize $\varepsilon(n)$ we proceed to determine the derivative of $\varepsilon(n)$ with respect to $w(n)$. The instantaneous gradient of the squared error can be express as

$$\frac{\partial}{\partial w} e^2(n) = \frac{\partial}{\partial w}[y(n) - w^T(n)x(n)]$$

$$= -2x(n)[y(n)-w^T(n)x(n)]$$

$$= -2x(n)e(n) \qquad (3.4)$$

where the gradient vector is defined as

$$\frac{\partial}{\partial w} = \left[\frac{\partial}{\partial w_0}, \frac{\partial}{\partial w_1}, \ldots, \frac{\partial}{\partial w_{M-1}}\right]^T$$

The LMS algorithm updates the filter coefficients based on the method of steepest descent. This can be described in vector notation as

$$w(n+1) = w(n) + \mu\left(-\frac{\partial E[e^2(n)]}{\partial w(n)}\right) \qquad (3.5)$$

where μ is the adaptation step size.

Substituting equation 3.4 into equation 3.5 yields

$$w(n+1) = w(n) + \mu(x(n)e(n)) \qquad (3.6)$$

where the factor of 2 in equation 3.4 has been absorbed in the adaptation step size μ.

It can be seen that the filter update equation is very simple. The LMS filter is widely used in adaptive filter applications such as adaptive equalization, echo cancellation etc..

3.1.3 LMS Algorithm for a M-order FIR Adaptive Filter

Inputs:	$x(n), y(n)$
Parameters :	M = Filter order
	μ = Step size
Initialization:	$w(0) = 0$
Computation:	For n = 0,1,2,...
	(a) $\quad (n) = w^T(n)x(n)$
	(b) $e(n) = y(n) - \quad (n)$
	(c) $w(n+1) = w(n) + \mu e(n)x(n)$

3.1.4 Convergence Rate

The convergence rate of the filter coefficients depends on the choice of the adaptation step size μ where $0 < \mu < 1/\lambda_{max}$. When the eigenvalues of the correlation matrix are unevenly spread, the filter coefficients converge at different speeds; the smaller the k^{th} eigenvalue the slower the speed of convergence of the k^{th} coefficients.

3.2 Recursive Least Squares (RLS) Adaptive Filters

3.2.1 Introduction

A recursive least squared error (RLS) filter is a sample adaptive, time-updated network. The RLS has a relatively fast rate of convergence to optimal filter coefficients. This is useful in applications such as speech enhancement, and echo cancellation, where the filter has to track the changes in the process.

Again, in the adaptive filtering method, we have to consider for the minimization of the mean-square error

$$\varepsilon(n) = E\{|e(n)|^2\}$$

3.2.2 Derivation of RLS

Let us assume that $x(n)$, $y(n)$ and $w(n)$ are the filter input, desired signal and the filter coefficient vectors respectively, where $w(n) = [w_0(n), w_1(n), \ldots, w_{M-1}(n)]$ and M is the order of the filter.

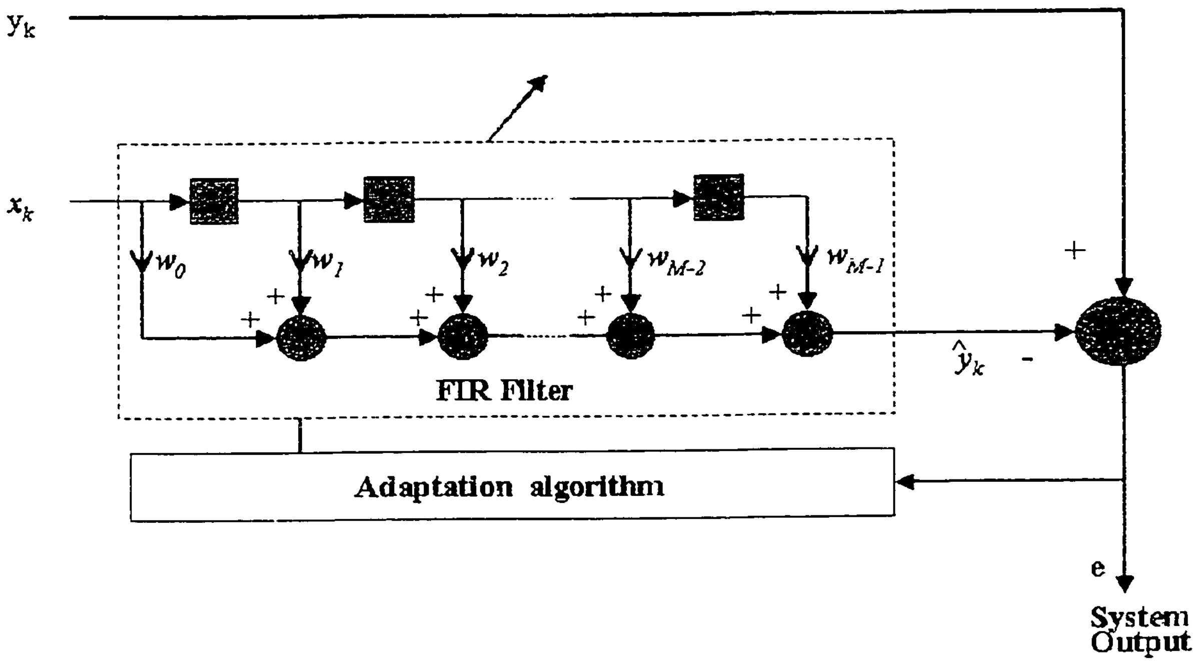

Figure 3.2 Illustration of the configuration of an adaptive filter

The filter output can be expressed as

$$\hat{y}(n) = w^T(n)x(n)$$

where $\hat{y}(n)$ is an estimate of the desired signal $y(n)$. The filter error signal is defined as

$$e(n) = y(n) - \hat{y}(n)$$
$$= y(n) - w^T(n)x(n)$$

The adaptation objective is to minimize the mean squared error defined as

$$\varepsilon(n) = E\{|e(n)|^2\}$$
$$= E\{[y(n) - w^T(n)x(n)]^2\}$$
$$= E[y^2(n)] - E[x(n)y(n)] + w^T(n)E[x(n)x^T(n)]w(n)$$
$$= r_{xx}(0) - 2w^T(n)r_{xy} + w^T(n)R_{xx}w(n)$$

where $r_{xy} = E[x(n)y(n)]$ is the cross correlation vector of the input and desired signals, and $R_{xx} = E[x(n)x^T(n)]$ is the autocorrelation matrix of the input signal

Thus, in minimizing $\varepsilon(n)$, to find the coefficients that minimize $\varepsilon(n)$ we proceed to evaluate the derivative of $\varepsilon(n)$ with respect to $w(n)$. Thus we have

$$\frac{\partial \varepsilon(n)}{\partial w} = -2r_{xy} + 2w^T R_{xx} \tag{3.6}$$

where the gradient vector is defined as

$$\frac{\partial}{\partial w} = \left[\frac{\partial}{\partial w_0}, \frac{\partial}{\partial w_1}, \ldots, \frac{\partial}{\partial w_{M-1}}\right]^T$$

The minimum of the mean squared error is obtained by setting $\frac{\partial}{\partial w}$ to zero as

$$0 = -2r_{xy} + 2w^T R_{xx}$$

$$\Rightarrow \quad \boldsymbol{R_{xx} w = r_{xy}} \qquad (3.7)$$

$$\Rightarrow \quad \boldsymbol{w} = R_{xx}^{-1} r_{xy} \qquad (3.8)$$

For a block of N sample vectors the correlation matrix can be written as

$$\boldsymbol{R_{xx} = X^T X = \sum_{n=0}^{N-1} x(n)x^T(n),} \qquad (3.9)$$

where $\boldsymbol{x(n)} = [x(n), x(n-1), \ldots, x(n-M+1)]^T$. Now the sum vector product in equation 3.9 can be expressed in a recursive fashion as

$$\boldsymbol{R_{xx}(n) = R_{xx}(n-1) + x(n)x^T(n)} \qquad (3.10)$$

To introduce adaptability to the time variations of the signal statistics, the autocorrelation estimate in equation 3.10 is modified to be

$$\boldsymbol{R_{xx}(n) = \lambda R_{xx}(n-1) + x(n)x^T(n)} \qquad (3.11)$$

Where $1 \geq \lambda > 0$.

Similarly the cross correlation vector is given by

$$r_{xy}(n) = \sum_{n=0}^{N-1} x(n)y(n) \tag{3.12}$$

The sum products in equation 3.12 can be calculated in recursive form as

$$r_{xy}(n) = r_{xy}(n\text{-}1) + x(n)y(n) \tag{3.13}$$

And again this equation can be made adaptive using an exponentially decaying forgetting factor λ as

$$r_{xy}(n) = \lambda r_{xy}(n\text{-}1) + x(n)y(n) \tag{3.14}$$

Since $R_{xx}(n)$ *and* $r_{xy}(n)$ both depend on n, instead of solving the deterministic normal equation directly, for each value of n, we will derive a recursive solution of the form

$$w_n = w_{n-1} + \Delta w_{n-1} \tag{3.15}$$

where Δw_{n-1} is a correlation that is applied to the solution at time n-1. Since

$$w = R_{xx}^{-1} r_{xy} \tag{3.16}$$

A recursive relation for the matrix inversion is obtained using the following lemma.

The Matrix Inversion Lemma : Let **A** and **B** be two positive-definite, $M \times M$ matrices related by

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^{T} \tag{3.17}$$

Where **D** is a positive-definite $N \times N$ matrix, **C** is $M \times N$ matrix. The matrix inversion lemma states that the inverse of the matrix A can be expressed as

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^{T}\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^{T}\mathbf{B} \tag{3.18}$$

This lemma can be proved by multiplying equation 3.17 and 3.18. The results of multiplication are the identity matrix in both the right hand side and left hand side.

The matrix inversion lemma can be used to obtain a recursive implementation for the inverse of the correlation matrix $R_{xx}^{-1}(n)$ as follows.

Let

$$R_{xx}(n) = \mathbf{A} \tag{3.19}$$

$$\lambda^{-1} R_{xx}^{-1}(n-1) = \mathbf{B} \tag{3.20}$$

$$\mathrm{x(n)} = \mathbf{C} \tag{3.21}$$

$$\mathrm{D} = \text{Identity Matrix} \tag{3.22}$$

Substituting equation 3.19, 3.20, 3.21, and 3,22 into 3.18. We obtain

$$R_{xx}^{-1}(n) = \lambda^{-1} R_{xx}^{-1}(n-1) - \frac{\lambda^{-2} R_{xx}^{-1}(n-1)x(n)x^T(n)R_{xx}^{-1}(n-1)}{1+\lambda^{-1}y^T(n)R_{xx}^{-1}(n-1)x(n)} \tag{3.23}$$

Now define the variable $\Phi(n)$ and $k(n)$ as

$$\Phi_{xx}(n) = R_{xx}^{-1}(n) \tag{3.24}$$

$$k(n) = \frac{\lambda^{-1} R_{xx}^{-1}(n-1)x(n)}{1+\lambda^{-1}y^T(n)R_{xx}^{-1}(n-1)x(n)} \tag{3.25}$$

or

$$k(n) = \frac{\lambda^{-1} \Phi_{xx}(n-1)x(n)}{1+\lambda^{-1}y^T(n)\Phi_{xx}(n-1)x(n)} \tag{3.26}$$

Substituting equation 3.24 and 3.26 into 3.23. we obtain

$$\Phi_{xx}(n) = \lambda^{-1}\Phi_{xx}(n-1) - \lambda^{-1}k(n)x^T(n)\Phi_{xx}(n-1) \tag{3.27}$$

from equation 3.26 and 3.27 we have

$$\begin{aligned} k(n) &= [\lambda^{-1}\Phi_{xx}(n-1) - \lambda^{-1}k(n)x^T(n)\Phi_{xx}(n-1)]x(n) \\ &= \Phi_{xx}(n)x(n) \end{aligned} \tag{3.28}$$

So the $\Phi_{xx}(n)$, $k(n)$ are used to derive the RLS adaptation algorithm.

Recursive Time-Update of Filter Coefficients: The least squared error filter coefficients are

$$w(n) = R_{xx}^{-1}(n)\,r_{xy}(n)$$

$$= \Phi_{xx}(n)r_{xy}(n) \qquad (3.29)$$

Substituting the recursive form of the correlation vector in equation 3.29

$$\Rightarrow \quad w(n) = \Phi_{xx}(n)[\,\lambda r_{xy}(n-1) + x(n)y(n)]$$

$$= \lambda\Phi_{xx}(n)r_{xy}(n-1) + \Phi_{xx}(n)x(n)y(n) \qquad (3.30)$$

And substituting the recursive form of $\Phi_{xx}(n)$ and $k(n) = \Phi_{xx}(n)x(n)$ into the right hand side of above equation

$$\Rightarrow \quad w(n) = [\lambda^{-1}\Phi_{xx}(n-1) - \lambda^{-1}k(n)x^T(n)\Phi_{xx}(n-1)]\lambda r_{xy}(n-1) + k(n)y(n) \quad (3.31)$$

or

$$\Rightarrow \quad w(n) = \Phi_{xx}(n-1)r_{xy}(n-1) - k(n)x^T(n)\Phi_{xx}(n-1)r_{xy}(n-1) + k(n)y(n) \quad (3.32)$$

Substituting $\Phi_{xx}(n-1)r_{xy}(n-1) = w(n-1)$ into above equation

$$\Rightarrow \quad w(n) = w(n-1) - k(n)[y(n) - x^T(n)w(n-1)] \qquad (3.33)$$

The equation can be rewritten in the following form

$$w(n) = w(n\text{-}1) - k(n)e(n) \qquad (3.34)$$

3.2.3 RLS Algorithm for a M-order FIR Adaptive Filter

Input Signals $\quad x(n),\ y(n)$

Initial values

$$\Phi_{xx}(0) = \delta I$$

$$w(0) = w_I$$

For n = 1,2,3....

Filter gain vector

$$k(n) = \frac{\lambda^{-1}\Phi_{xx}(n-1)x(n)}{1+\lambda^{-1}y^{T}(n)\Phi_{xx}(n-1)x(n)}$$

Error signal equation

$$e(n) = y(n) - w^{T}(n\text{-}1)x(n)$$

Filter coefficients

$$w(n) = w(n-1) - k(n)e(n)$$

Inverse correlation matrix update

$$\Phi_{xx}(n) = \lambda^{-1}\,\Phi_{xx}(n-1) - \lambda^{-1}\,k(n)x^{T}(n)\;\Phi_{xx}(n-1)$$

3.2.4 Learning Curve of the RLS Algorithm

In the RLS algorithm there are two errors, the prior estimation error $\varepsilon(n)$ and the a posteriori estimation error $e(n)$, to be considered. Given the initial condition previous section we find that the mean-square values of these two errors vary differently with time n. At time n=1, the mean-square value of $\varepsilon(n)$ attains a large value, equal to the mean-square value of the desired response $y(n)$, and then decays with increasing n. The mean-square value $e(n)$, on the other hand, attains a small value at n=1, and then rises with increasing n. Accordingly, the choice of $\varepsilon(n)$ as the error of interest yields a learning curve for the RLS algorithm that has the same general shape as that for the LMS algorithm. By so doing, we can then make a direct graphical comparison between the learning curves of the RLS and LMS algorithms. We will therefore base computation of the ensemble-averaged learning curve of the RLS algorithm on a prior estimation error $\varepsilon(n)$.

We may express the a prior estimate $\varepsilon(n)$ as

$$\varepsilon(n) = e_0(n) - [w(n-1) - w_0(n)]x(n)$$

$$= e_0(n) - \xi(n-1)x(n) \qquad (3.35)$$

where $\xi(n-1)$ is the weight-error vector at time $n-1$. As an index of statistical performance for the RLS algorithm, it is convenient to use a prior estimate error $\varepsilon(n)$ to define the mean-square error:

$$J'(n) = E[|\varepsilon(n)|^2] \qquad (3.36)$$

To prime the symbol $J'(n)$ is intended to distinguish the mean-square value of $\varepsilon(n)$ from that of $e(n)$. Substituting Eq. 3.35 in 3.36, and then expanding terms, we get

$$J'(n) = E[|e_0(n)|^2] + E[x(n)\xi(n-1)x(n)]$$

$$- E[\xi(n-1)x(n)e_0^*(n)] - E[e_0(n)x(n)\ \xi(n-1)] \qquad (3.37)$$

With the measurement $e_0(n)$ assumed to be of zero mean, the first expectation on the right-hand side of Eq. 3.37 is simply the variance of $e_0(n)$, which is denoted by σ^2. As for the remaining three expectations, we may make the following observations in light of the independence assumption described previously:

1. The estimate $w(n-1)$, and therefore the weight-error vector $\xi(n-1)$, is independent of the tap-input vector $x(n)$; the latter is assumed to be drawn from a wide-sense stationary process of zero mean. Hence, we may use this statistical independence together with well-known results from matrix algebra to express the second expectation on the right-hand side of Eq. 3.37 as follows:

$$
\begin{aligned}
E[x^T(n)\xi(n-1)\xi^T(n-1)x(n)] &= E[tr\{x^T(n)\xi(n-1)\xi^T(n-1)x(n)\}] \\
&= E[tr\{x(n)x^T(n)\xi(n-1)\xi^T(n-1)\}] \\
&= tr\{E[x(n)x^T(n)\xi(n-1)\xi^T(n-1)]\} \\
&= tr\{E[x(n)x^T(n)]E[\xi(n-1)\xi^T(n-1)]\} \\
&= tr\{\boldsymbol{RK}(n-1)\}, \qquad (3.38)
\end{aligned}
$$

where, in the last line, we have made use of the definitions of the ensemble-averaged correlation matrix $\boldsymbol{R}$ and weight-error correlation matrix $\boldsymbol{K}(n-1)$.

2. The measurement error $e_0(n)$ depends on the tap-input vector $x(n)$; this follows from a simple rearrangement. The weight-error $\xi(n-1)$ is therefore independent of both $x(n)$ and $e_0(n)$. Accordingly, we may show that the third expectation on the right-hand side of Eq. 3.37 is zero by first reformulating it as follows:

$$E[\xi^T(n-1)x(n)e_0^*(n)] = E[\xi^T(n-1)]E[x(n)e_0^*(n)]$$

We now recognize from the principle of orthogonality that all the elements of the tap-input vector $x(n)$ are orthogonal to the measurement error $e_0(n)$. We therefore have

$$E[\xi^T(n-1)x(n)e_0{}^*(n)] = 0 \qquad (3.39)$$

3. The fourth and final expectation on the right-hand side of Eq. 3.37 has the same mathematical form as that just considered in point 2, except for a trivial complex conjugation. We may therefore set this expectation equal to zero, too:

$$E[e_0(n)x^T(n)\xi(n-1)] = 0 \qquad (3.40)$$

Thus, recognizing that $E[|e_0(n)|^2] = \sigma^2$, and using the results of Eq. 3.38 to 3.39 in 3.37, we get the following simple formula for the mean-squared error in the RLS algorithm.

$$J'(n) = \sigma^2 + tr[\mathbf{RK}(n-1)] \qquad (3.41)$$

Where $K(n) = \dfrac{\sigma^2}{n-M-1}R^{-1}$.

Next, substituting $K(n)$ in Eq. 3.41, we get (for $\lambda = 1$)

$$J'(n) = \sigma^2 + \frac{M\sigma^2}{n-M-1}, \qquad n > M+1 \qquad (3.42)$$

Base on this result, we may make the following deductions [7]:

- The ensemble-averaged learning curve of the RLS algorithm converges in about 2M iterations, where M is the number of taps in the transversal filter.

This means that the rate of convergence of the RLS algorithm is typically an order of magnitude faster than that of the LMS algorithm.

- As the number of iteration, n, approaches infinity, the mean-squared error $J'(n)$ approaches a final value equal to the variance σ^2 of the measurement error $e_0(n)$. In other words the RLS algorithm, in theory, produces zero excess mean-squared error (or, equivalently, zero misadjustment) when operating in a stationary environment.
- Convergence of the RLS algorithm in the mean square is independent of the eigenvalues of the ensemble-averaged correlation matrix **R** of the input vector $x(n)$.

It should be emphasized that the above-mentioned improvement in the rate of convergence of the RLS algorithm over the LMS algorithm holds only when the measurement error $e_0(n)$ is small compared to the desired response $y(n)$, that is, when the signal-to-noise ratio is high. Also, the zero misadjustment property of the RLS algorithm assumes that the exponential weighting factor λ equals unity; that is, the algorithm operates with infinite memory.

Chapter 4

4 Normalized LMS Algorithm

In the standard form of LMS algorithm, the correction $\mu*x(n)*e(n)$ applied to the tap-weight vector $w(n)$ at iteration $n+1$ is directly proportional to the tap-input vector $x(n)$. Therefore, when $x(n)$ is large, the LMS algorithm experiences a gradient noise amplification problem. To overcome this difficulty, we may use the normalized LMS algorithm.

4.1 Normalized LMS Algorithm as the Solution to a Constrained Optimization Problem

The normalized LMS algorithm may be viewed as the solution to a constrained optimization (minimization) problem. Specifically, the problem of interest may be stated as follows:

Given the tap-input vector $x(n)$ and the desired response $y(n)$, determine the tap-weight vector $w(n+1)$ so as to minimize the squared Euclidean norm of the change

$$\delta w(n+1) = w(n+1) - w(n) \tag{4.1}$$

in the tap-weight vector $w(n+1)$ with respect to its old value $w(n)$, subject to the constraint

$$w(n+1)x(n) = y(n) \tag{4.2}$$

To solve this constrained optimization problem, we may use the method of Lagrange multipliers.

The squared norm of the change $\delta w(n+1)$ in the tap-weight vector $w(n+1)$ may be expressed as

$$\begin{aligned} ||\delta w(n+1)||^2 &= \delta w(n+1)\ \delta w(n+1) \\ &= [w(n+1) - w(n)][\ w(n+1) - w(n)] \\ &= \sum_{k=0}^{M-1} |\ w_k(n+1) - w_k(n)\ |^2 \end{aligned} \tag{4.3}$$

Define the tap weight $w(n)$ for $k = 0,1,2,\ldots,$ M-1 in terms of its real and imaginary parts by writing

$$w_k(n) = a_k(n) + jb_k(n), \qquad k = 0,1,2,\ldots, M-1 \qquad (4.4)$$

we then have

$$||\delta w(n+1)||^2 = \sum_{k=0}^{M-1}([a_k(n+1) - a_k(n)]^2 + [b_k(n+1) - b_k(n)]^2) \qquad (4.5)$$

Let the tap input $x(n\text{-}k)$ and the desired response $y(n)$ be defined in terms of their respective real and imaginary parts as follows:

$$y(n) = y_1(n) + jy_2(n) \qquad (4.6)$$

$$x(n\text{-}k) = x_1(n\text{-}k) + jx_2(n\text{-}k) \qquad (4.7)$$

Accordingly, we may rewrite the complex constraint of Eq. 4.2 as an equivalent pair of real constraints:

$$\sum_{k=0}^{M-1}(a_k(n+1)x_1(n-k) + b_k(n+1)x_2(n-k)) = y_1(n) \qquad (4.8)$$

and

$$\sum_{k=0}^{M-1}(a_k(n+1)x_2(n-k) - b_k(n+1)x_1(n-k)) = y_2(n) \qquad (4.9)$$

we are now ready to formulate a real-valued cost function J(n) for the constrained optimization problem at hand. In particular, we combine Eqs. 4.5, 4.8 and 4.9 into a single relation:

$$J(n) = \sum_{k=0}^{M-1}([a_k(n+1) - a_k(n)]^2 + [b_k(n+1) - b_k(n)]^2)$$

$$+ \lambda_1\left[y_1 - \sum_{k=0}^{M-1}(a_k(n+1)x_1(n-k) + b_k(n+1)x_2(n-k))\right]$$

$$+ \lambda_2\left[y_2 - \sum_{k=0}^{M-1}(a_k(n+1)x_2(n-k) - b_k(n+1)x_1(n-k))\right] \quad (4.10)$$

where λ_1 and λ_2 are Lagrange multipliers. To find the optimum values of $a_k(n+1)$ and $b_k(n+1)$, we differentiate the cost function $J(n)$ with respect to those two parameters and then set the results equal to zero. Hence, the use of Eq. 4.10 in the equation

$$\frac{\partial J(n)}{\partial a_k(n+1)} = 0$$

yields the result

$$2[a_k(n+1) - a_k(n)] - \lambda_1 x_1(n-k) - \lambda_2 x_2(n-k) = 0 \quad (4.11)$$

Similarly, the use of Eq. 4.10 in the complementary equation

$$\frac{\partial J(n)}{\partial b_k(n+1)} = 0$$

yields the complementary result

$$2[b_k(n+1) - b_k(n)] - \lambda_1 x_2(n-k) + \lambda_2 x_1(n-k) = 0 \quad (4.12)$$

Next, we use the definitions of Eqs. 4.4 and 4.7 to combine these two real results into a single complex one, as shown by

$$2[w_k(n+1) - w_k(n)] = \lambda^* x(n-k), \quad k = 0,1,2,.., M-1 \tag{4.13}$$

where λ is a complex Lagrange multiplier:

$$\lambda = \lambda_1 + j\lambda_2 \tag{4.14}$$

To solve for the unknown λ^*, we multiply both sides of Eq. 4.13 by $x^*(n-k)$ and then sum over all possible integer values of k for 0 to M-1. We thus get

$$\lambda^* = \frac{2}{\sum_{k=0}^{M-1} |x(n-k)|^2} \left[\sum_{k=0}^{M-1} w_k(n+1)x^*(n-k) - \sum_{k=0}^{M-1} w_k(n)x^*(n-k) \right]$$

$$= \frac{2}{\| x(n) \|^2} [w^T(n+1)x^*(n) - w^T(n)x^*(n)] \tag{4.15}$$

where $||x(n)||$ is the Euclidean norm of the tap-input vector $x(n)$. Next, we use the complex constraint of Eq. 4.15 and thus formulate λ^* as follows:

$$\lambda^* = \frac{2}{\| x(n) \|^2} [d^*(n) - w^T(n)x^*(n)] \tag{4.16}$$

However, from the definition of the estimation error e(n), we have

$$e(n) = d(n) - w(n)x(n)$$

Accordingly, we may further simplify the expression given in Eq. 4.16 and thus write

$$\lambda^* = \frac{2}{\| x(n) \|^2} e^*(n) \tag{4.17}$$

Finally, we substitute Eq. 4.17 into Eq. 4.13, obtaining

$$\begin{aligned} \delta w_k(n+1) &= w_k(n+1) - w_k(n) \\ &= \frac{1}{\| x(n) \|^2} x(n-k)e^*(n), \qquad k = 0, 1, \ldots, M-1 \end{aligned} \tag{4.18}$$

In vector form, we may equivalently write

$$\begin{aligned} \delta w(n+1) &= w(n+1) - w(n) \\ &= \frac{1}{\| x(n) \|^2} x(n)e^*(n) \end{aligned} \tag{4.19}$$

In order to exercise control over the change in the tap-weight vector from one iteration to the next without changing its direction, the real scaling factor μ is used. Redefining the change $\delta w(n+1)$ simply as

$$\delta w(n+1) = w(n+1) - w(n)$$

$$= \frac{\mu}{\| x(n) \|^2} x(n)e^*(n) \tag{4.20}$$

Equivalently, we may write

$$w(n+1) = w(n) + \frac{\mu}{\| x(n) \|^2} x(n)e^*(n) \tag{4.21}$$

Indeed, this is the desired recursive for computing the M-by-1 tap-weight vector in the normalized LMS algorithm.

Equation 4.21 clearly shows the reason for using the term "normalized". In particular, we see that the product vector $x(n)e^*(n)$ is normalized with respect to the squared Euclidean norm of the tap-input vector $x(n)$.

The important point to note from the analysis presented above is that given new input data (at time n) represented by the tap-input vector $x(n)$ and desired response $d(n)$, the normalized LMS algorithm updates the tap-weight vector in such a way that the value $w(n+1)$ computed at time $n+1$ exhibits the minimum change (in a Euclidean norm sense) with respect to the known value $w(n)$ at time n; for example, no charge may represent minimum change. Hence, the normalized LMS algorithm (and for that matter the conventional LMS algorithm) is a manifestation of the principle of minimal disturbance.

The principle of minimal disturbance states that, in the light of new input data, the parameters of an adaptive system should only be disturbed in a minimal fashion.

Moreover, comparing the recursion of Eq. 4.21 for the normalized LMS algorithm with that of Eq. 3.6 for the conventional LMS algorithm, we may make the following observations: [7]

- The adaptation constant μ for the normalized LMS algorithm is dimensionless whereas the adaptation constant μ for the LMS algorithm has the dimensions of inverse power.
- Setting

$$u(n) = \frac{\mu}{\| x(n) \|^2} \tag{4.22}$$

  we may view the normalized LMS algorithm as an LMS algorithm with a time-varying step-size parameters.
- The normalized LMS algorithm is convergent in the mean square if the adaptation constant μ satisfies the following condition:

$$0 < \mu < 2 \tag{4.23}$$

Most importantly, the normalized LMS algorithm exhibits a rate of convergence that is potentially faster than that of the standard LMS algorithm for both uncorrelated and correlated input data. Another point of interest is that in overcoming the gradient noise amplification problem associated with the LMS algorithm, the normalized algorithm introduces a problem of its own. Specifically, when the tap-input vector *x(n)* is small, numerical difficulties may arise because then we have to divide by a small value for the squared norm $||x(n)||^2$. To overcome this problem, we slightly modify the recursion of Eq. 4.21 as follows:

$$w(n+1) = w(n) + \frac{\mu}{a + ||x(n)||^2} x(n) e^*(n) \tag{4.23}$$

where $a > 0$, and as before $0 < \mu < 2$.

4.2 Summary of Normalized LMS Algorithm

Parameters: M = number of taps (filter order)

μ = adaptation constant

$0 < \mu < 2$

a = positive constant

Initialization: if prior knowledge on the tap-weight vector w(n) is available, use it to select an appropriate value for w(0). Otherwise, set w(0) = 0.

Data:

(a) Given: x(n): M-by-1 tap-input vector at time n

(b) To be computed: w(n+1) = estimate of tap-weight vector at time n+1

Computation:

$$n = 0, 1, 2, \ldots$$

$$e(n) = d(n) - w(n)x(n)$$

$$w(n+1) = w(n) + \frac{\mu}{a + \| x(n) \|^2} x(n)e^*(n)$$

Chapter 5

5 Acoustic Echo Cancellation and Noise Reduction System

5.1 Introduction

The problem of combined acoustic echo cancellation and noise reduction has found considerable interest recently. The interest is fueled by application in mobile communications where both acoustic echo cancellation and noise reduction are necessary to achieve sufficient quality of the transmitted speech signal. The realization of such a combined system is a challenging task. The difficulties of acoustic echo cancellation are mainly due to the high computational complexity of the echo canceller and influences which disturb the adaptation of the canceller such as ambient noise, near end speech, and variation of the acoustic environment.

The noise reduction task is also not easily solved since in the typical reverberant environment no "noise only" reference signal can be obtained which is sufficiently correlated to the noise within the microphone signal. A powerful hands-free speech transmission system should combine both NR and an AEC. Acoustic echo cancellation and noise reduction using two adaptive filters are effective techniques of reducing noise and acoustic cancellation in a hands-free telephone system. The traditional acoustic echo cancellation using adaptive finite impulse response (FIR), the traditional noise reduction by using FIR and the combination of these two FIR filters in order to deal with four operation modes in a hands-free phone conversation or voice command are presented. The four modes: idle (IDLE), receive mode (RM), transmit mode (TM), and double-talk mode (DT) will be discussed later.

5.2 Acoustic Echo Cancellation Adaptive FIR Filter

5.2.1 Acoustic Echo Generation

In hands-free telephony (teleconferencing), echoes occur because an open-air acoustic path exists between the loudspeaker and microphone. Speech originating in either room is transmitted over the telephone network to the other room where it is amplified and reproduced by an audio loudspeaker (see Figure 5.1). The output of the loudspeaker fills the entirety of that room and, via many paths of reflections, reaches the microphone in that room. This acoustically echoed speech signal is transmitted back over the network to the originating room and is reproduced by the loudspeaker where it is perceived as an echo.

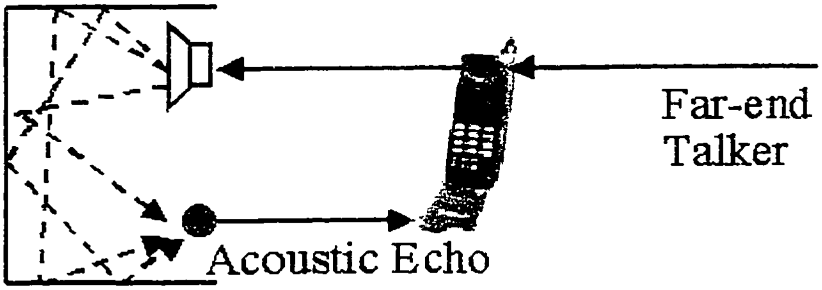

Figure 5.1 Acoustic Echo

Acoustic echoes degrade the quality of speech communications in two ways. [10] First, echoes of speech are subjectively annoying to the person speaking. In fact, if the elapsed time between when a word is spoken and when its echo is heard is more than 300ms, the echo will actually cause most people to stutter. Second, echoes can overload communication circuits, resulting in a feedback condition called howling. If, by the combination of loudspeaker volume and microphone sensitivity, the echoes are louder than the originating speech, the teleconferencing equipment or the network itself can overload.

Reverberation is a related problem that is also caused by acoustic echoes. Reverberation occurs when the local audio signal is picked up by the microphone due to multiple signal paths within the room. When the local person speaks, the audio signal travels through multiple paths to reach the microphone. The primary path is directly from the person's mouth to the microphone, but there are alternative routes for the audio signal to travel, such as reflecting off a wall or table surface and then into the microphone. In a "reverberant room," there may be multiple reflections around the room in addition to the direct coupling. The effect of reverberation on the signal is to make the signal seem hollow and resonant. This effect is often described as making the audio sound "muddy."

5.2.2 Acoustic Echo Cancellation System

One of the most important issues in interactive communication is the quality of the audio. In videoconferencing systems, even if the video quality is comparatively good, poor audio will negatively impact the perceived quality of the video. In order for interactive communication to occur between two or more parties, the communication should be as close as possible to the face-to-face experiences.

To cancel echoes, the AEC must learn the character of the open-air path between the loudspeaker and microphone. This path is a function of not only the loudspeaker and microphone, but also of their placement within the room and the room's acoustics, including its construction materials, dimensions, furnishings and their locations, and the room's occupants.

The traditional acoustic echo cancellation (AEC) is illustrated in Figure 5.2

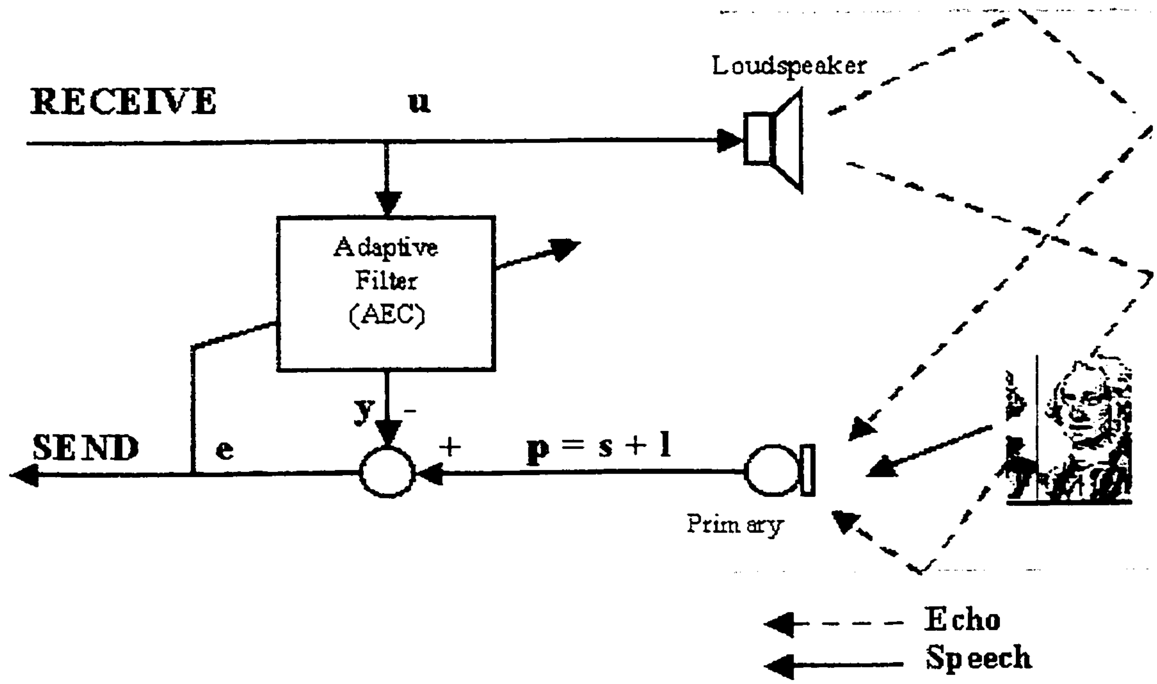

Figure 5.2 Acoustic Echo Cancellation System

In the echo cancellation system, the echo is most likely caused by the loudspeaker, which is amplifying the far-end talker's speech. The loudspeaker's output is reflecting from the walls, ceiling, or other object in an enclosed environment. From the Figure 5.1, u and p are denoted as far-end signal and primary microphone signal respectively. S is unknown speech signal and l is also an unknown echo caused by the loudspeaker and reflecting surfaces. The filter task is to use far-end signal u to produce an output y which is a close replica of the echo l. The output of the filter is subtracted from the primary input $p = s + l$ to produce the system output $e = p - y$.

5.3 Noise Reduction Adaptive FIR Filter

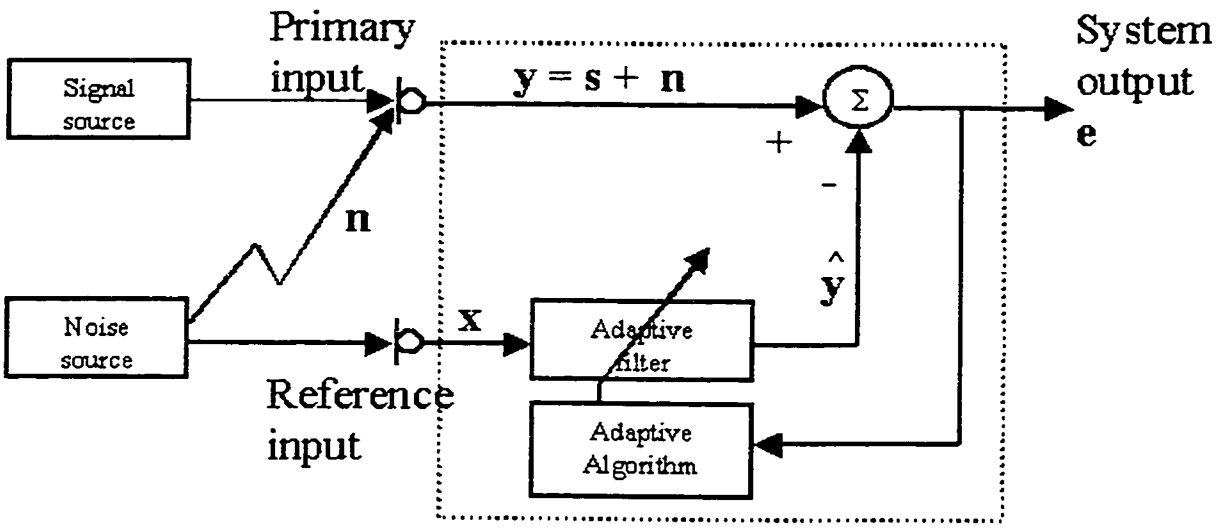

Figure 5.3 Noise Reduction Adaptive Filter System

A signal s is transmitted over a channel to a sensor that receives the desiredsignal plus an uncorrelated noise signal, n. The combined signal and noise, the corrupted signal $y= s + n$, form the "primary input " to the canceller. A second sensor receives a noise signal x which is uncorrelated with the signal but correlated in some unknown way with the noise n. The second sensor provides the "reference input" to the canceller. The noise x is filtered to produce an output that is a close replica of n. This output is subtracted from the primary input $y = s + n$ to produce the system output $e = y - \hat{y}$.

5.4 The Principle of Two Adaptive Filters AEC and NR

5.4.1 Introduction

Four operation modes, idle (IDLE), receive (RM), transmit (TM) and double-talk (DT) are identified to effectively cancel the acoustic echo and noise reduction in the primary signal. The system has these four states and each of them has a different signal, noise or echo present in either primary or reference microphones. Adaptive filters will be used in each of these four states. The contents of microphones in each state are shown below:

1. IDLE
   - Primary microphone: only noise is present, no speech.
   - Reference microphone: only nose is present.
2. Receive
   - Primary microphone: noise and echo are present, no speech.
   - Reference microphone: noise and echo are present.
3. Transmit
   - Primary microphone: both noise and near end speech are present.
   - Reference microphone: noise only.
4. Double-talk
   - Primary microphone: noise, echo, and near-end speech are present.
   - Reference microphone: both noise and echo are present.

5.4.2 IDLE Mode

The processing system IDLE mode is shown in Figure 5.4, where only background noise from the noise source is present. In the figure, *n* and *x* are outputs from the primary and reference microphones respectively.

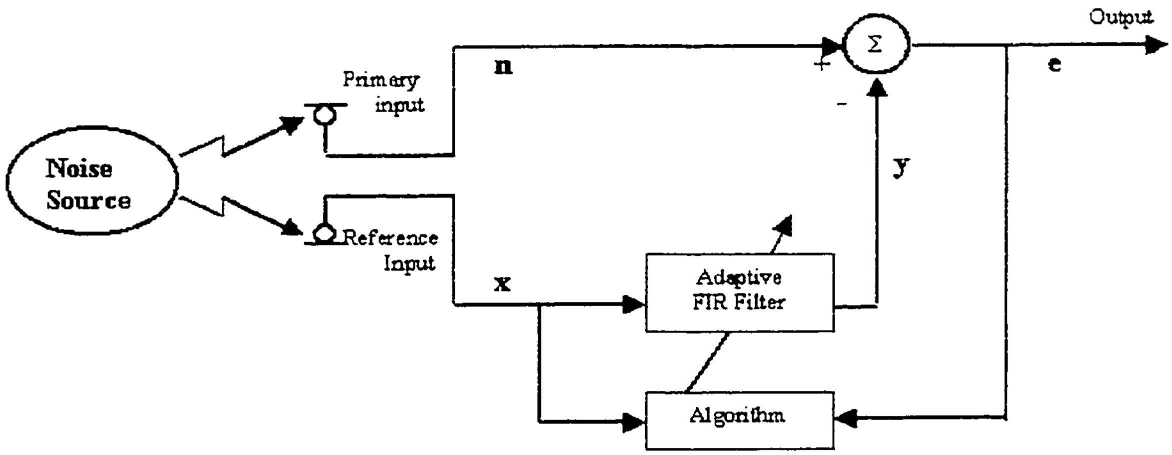

Figure 5.4 Principle of the Processing System in IDLE Mode

Output $\quad e = n - y$

$$= n - wx$$

where n is the primary microphone input

w is the tap-weights of the filter

x is the reference microphone input

y is the result of adaptive filter

5.4.3 Receive Mode

The processing system in RM mode is shown in Figure 5.5, where the far-end speech from the loudspeaker and the background noise from the noise source are present

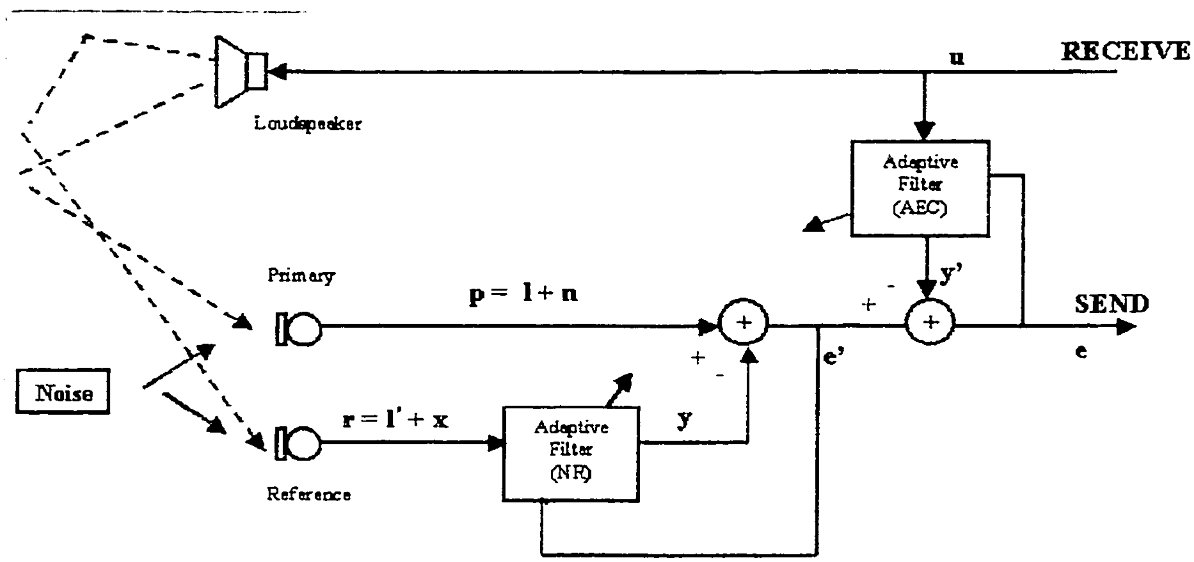

Figure 5.5 Principle of the Processing System in Receive Mode

Output $\quad e = p - y - y'$

$$= p - w \cdot r - w' \cdot u$$

where p is the primary microphone input

r is the reference microphone input

u is the far-end speech

w is the tap-weight at NR

w' is the tap-weight at AEC

5.4.4 Transmit Mode

The processing system in transmit mode is shown in Figure 5.6, where the near-end speech and the background noise are present.

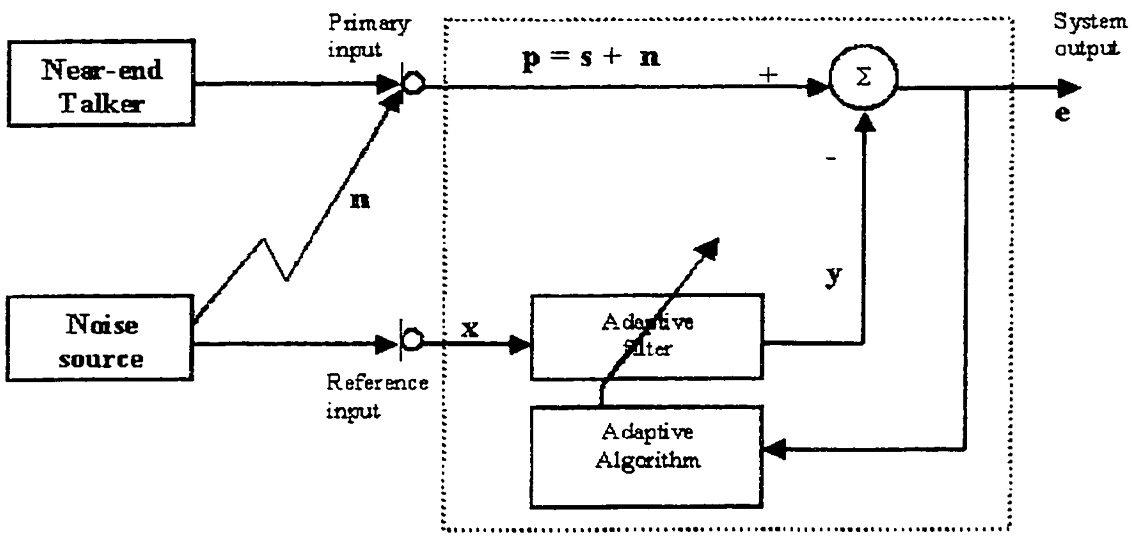

Figure 5.6 Principle of the Processing System in Transmit Mode

Output $\quad e = p - y$

$$= p - wx$$

where p is the primary microphone input

w is the tap-weights of the filter

x is the reference microphone input

y is the result of adaptive filter

5.4.5 Double-Talk Mode

The processing system in double-talk mode is similar to the RM figure and is shown in Figure 5.7 with presence of near-end speech, far-end speech (echo), and background noise.

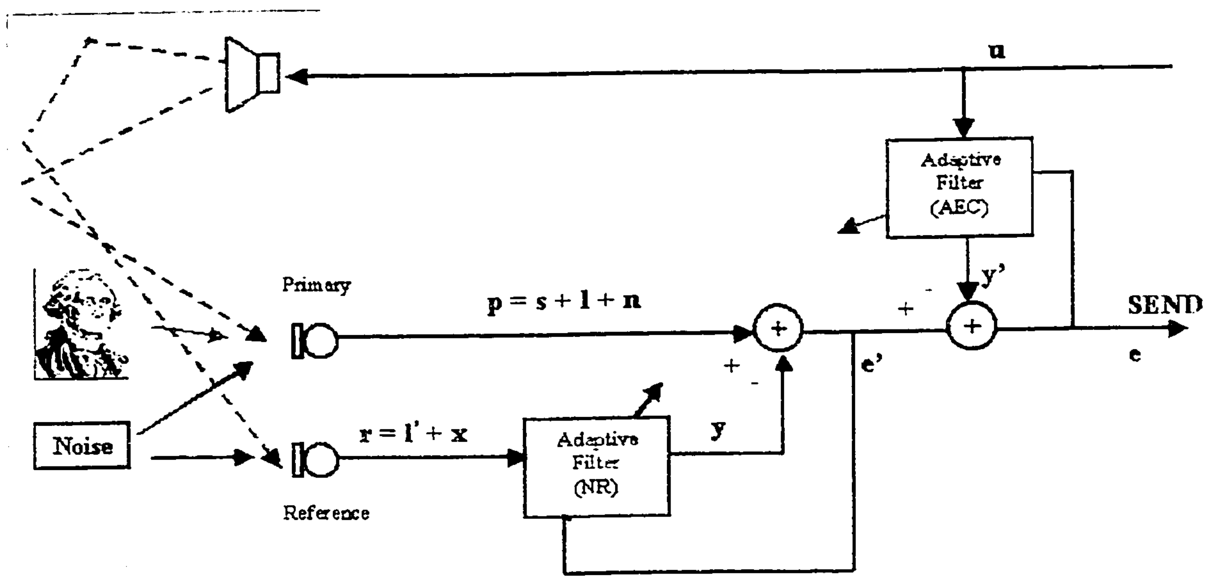

Figure 5.7 Principle of the processing system in Double-talk mode

Output $\quad e = p - y - y'$

$$= p - w \cdot r - w' \cdot u$$

where p is the primary microphone input

r is the reference microphone input

u is the far-end speech

w is the tap-weight at NR

w' is the tap-weight at AEC

Chapter 6

6 Experiment and Results

6.1 Database Collection

To test the adaptive FIR filter systems in operation modes, we need to have some sound files which contain many different situations.

The pre-recorded data that we used for the simulation are designed to test in different situations. The collection of speech, noise, echo files are accessible in computer form (wav files). The following is the description of the data files:

wav files contents:

- Near end speech:

  Is a 30 seconds duration female's voice, with 6 words.

- Far end speech:

  Is again a 30 second duration male's voice with almost nonstop talking in this duration.

Background Noise

	known	unknown
stationary	Engine etc.	Road, wind, air-conditioner etc
non-stationary	Car stereo speaker out, navigation guide, traffic information guide etc.	Bump, wiper, winker, conversation, noise when passing a car running to opposite direction

Table 6.1 Additive Noises in a Car Cabin

Noise and echo recording condition:

- In a car, BMW 325i 1994 with sunroof.
- Car radio off, air-condition off.
- Driver side's window open.
- In a regular street, car moving speed is around 50 Km/h.
- Signal-channel.

Recording equipment and software:

- Microphones : Lab-tec AM-242
- Cool-Edit 2000
- Acer 602 TER

Sampling rate:

- 16 kHz, 16 Bits, Mono wav format.

Wav file format

The WAVE file format is a subset of Microsoft's RIFF spec, which can include lots of different kinds of data. It was originally intended for multimedia files, but the spec is open enough to allow pretty much anything to be placed in such a file, and ignored by programs that read the format correctly.

Table 6.2 Structure of wav files

Structure of wav files	
4 characters:	RIFF
4-byte integer:	{length of sound array (in bytes) + length of header (36 bytes)}
4 characters:	WAVE
4 characters:	fmt{space}
4-byte integer:	{data word size - 8 or 16 bits}
2-byte integer:	{coding format - 1 }
2-byte integer:	{number channels - 1 (mono) or 2 (stereo)}
4-byte integer:	{samples per second e.g. 44100}
4-byte integer:	{bytes per second e.g. 176400}
2-byte integer:	{block align - data word size x number channels e.g.16 x 2}
2-byte integer:	{bits per sample e.g. 16}
4 characters:	data
4-byte integer:	{length of sound array (in bytes)}

6.2 Experiment Design

Three kinds of experiments were designed to test the adaptive interference filter system: acoustic echo cancellation only, noise reduction only and combination of acoustic echo cancellation and noise reduction. And two sets of data were tested in all these three systems. Each set of data consist of two kinds of speech and background noise, male speech and female speech are far-end speech and near end speech respectively. Acoustic echo cancellation system consists of speech signal corrupted by echo, noise reduction system consists of speech signal corrupted by background noise and the combination of AEC and NR system consists of speech signal corrupted by echo and noise. Apparently, the combination of two filters is dealing with more complicated corrupted speech signal. The echoes only or noise only system is showing better results than the two filters system, because each of them are dealing with less complicated corrupted signal. But the result is shown that the two filters system performs an acceptable result as well.

As mentioned before, two sets of data were used in the project, 6.5 seconds and 30 seconds in duration.

6.3 Results and Discussion

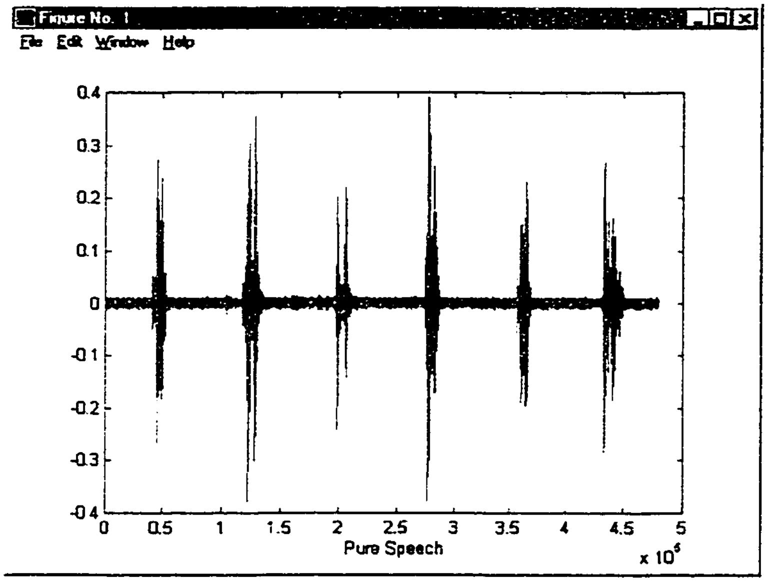

Figure 6.1 Speech signal

The Non-Corrupted Speech Signal is a female voice with a duration of 30 second. The purpose of experiments is to find the output of each system, either AEC, NR or combination of AEC and NR, as close to the pure speech signal as possible.

The follow pages will show the output figures of each system in LMS, RLS and NLMS. For comparison purpose of input and outputs, a corrupted signal will be shown in each page too.

6.3.1 Performance of Acoustic Echo cancellation

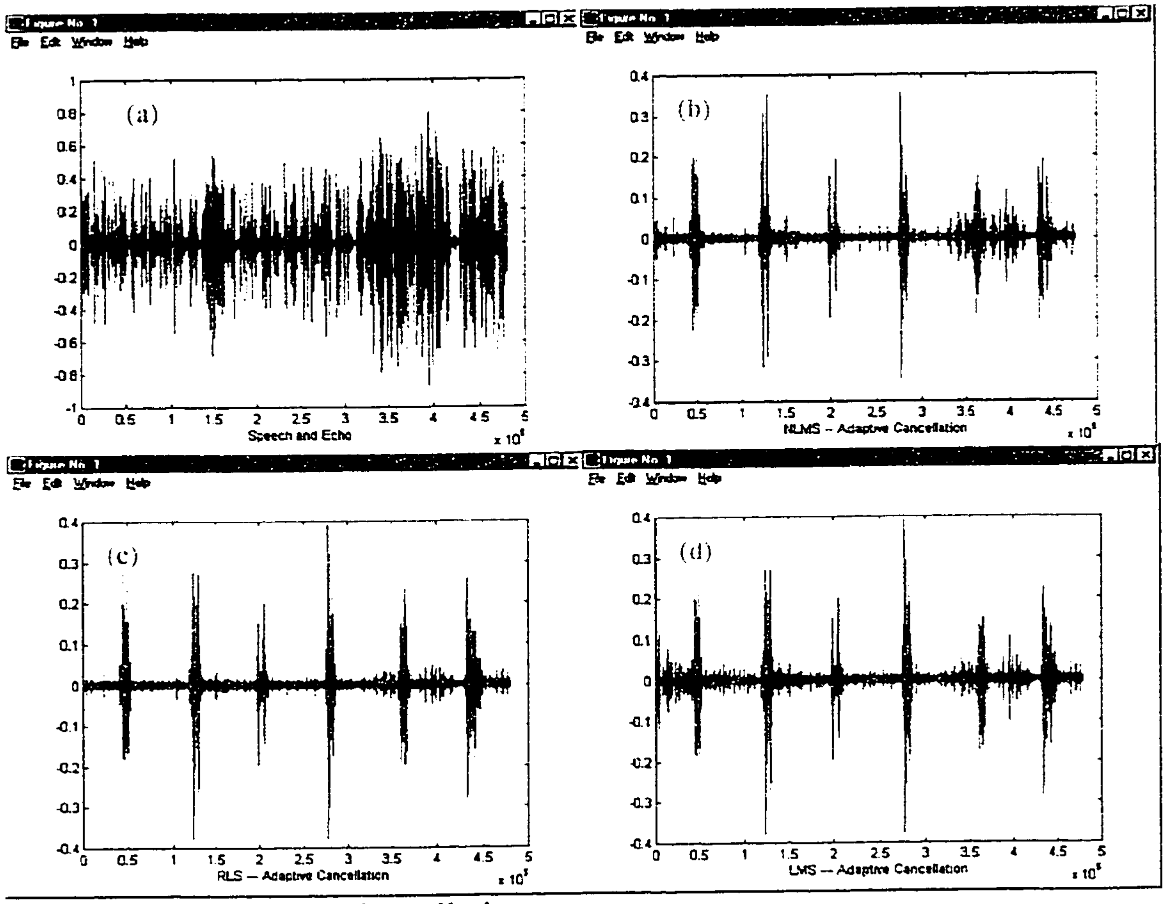

Figure 6.2 Acoustic Echo Cancellation

Priamry : Speech + echo (caused by Far-end speech)

Reference: Far-end speech

Figure 6.2 (a) is the primary microphone signal, (b) is the output signal of NLMS algorithm, (c) is the output signal of RLS algorithm and (d) is the output signal of LMS algorithm.

Apparently, the output of the RLS and NLMS algorithm are producing a better speech quality than LMS algorithm. The convergence of the LMS is easily to be shown that much slower than the RLS algorithm. The difference between RLS and NLMS is also the convergence speed, although NLMS convergence speed is faster than LMS. On the other hand, the computation complexity of RLS caused the time consumption in program running time.

6.3.2 Performance of Four States

There are four states in a hands-free system, Idle, Receive Mode, Transmit Mode, and Double-Talk mode. The two most critical states of the system are receive and double-talk. Both echo and noise are present in both primary and reference microphones. Dealing with the noise and echo at a same time is always a big challenge to achieve.

1. IDLE Mode

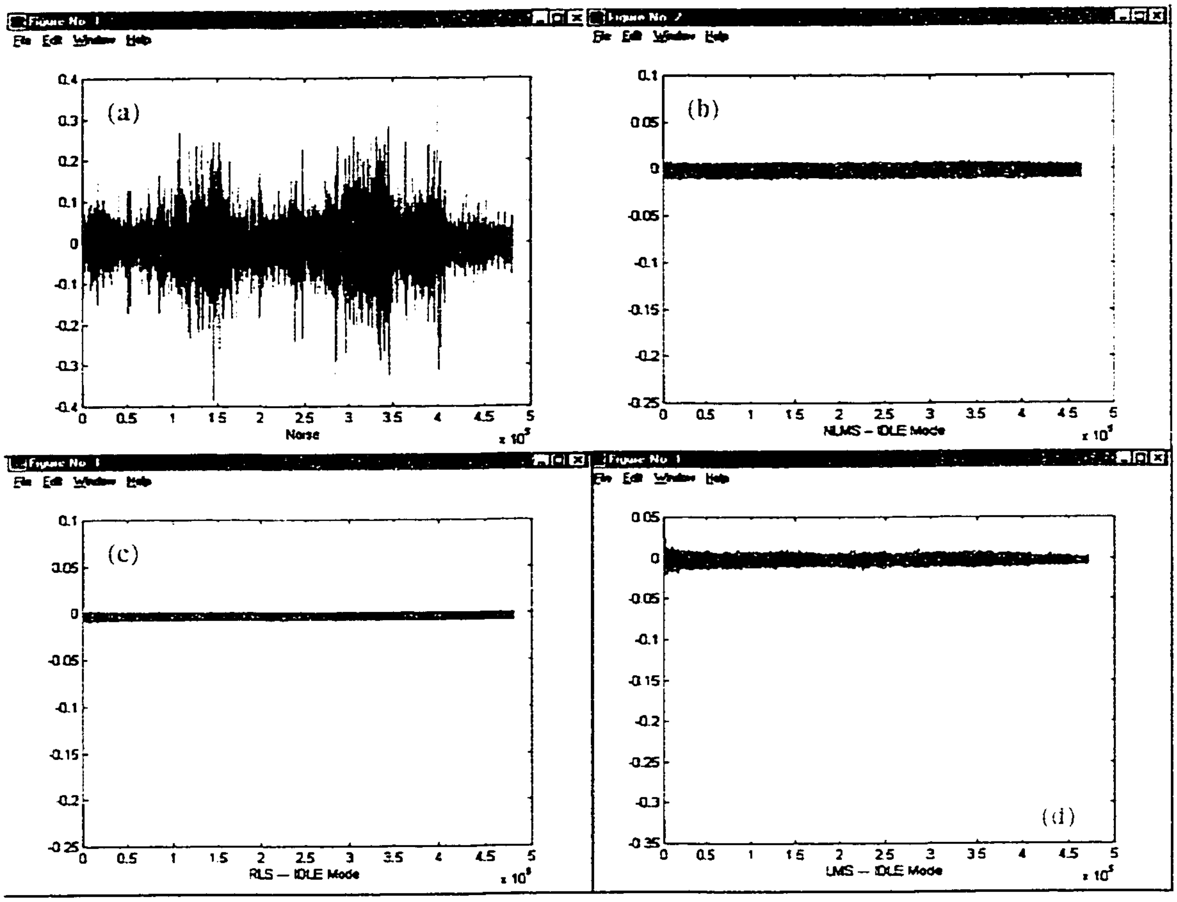

Figure 6.3 IDLE Mode

- Primary: noise.
- Reference: noise.

The Figure 6.3 (a), (b) , (c), and (d) are primary input, NLMS output, RLS output and LMS output respectively. The ideal output should be close to silent in IDLE mode, the figures have shown that the NLMS, LMS and RLS algorithms have performed a reasonable quality of output. Especially the output of RLS is pretty much close to silient.

And NLMS is also clearly shown that it is better than LMS in terms of convergence speed and performance.

2. **Receive Mode**

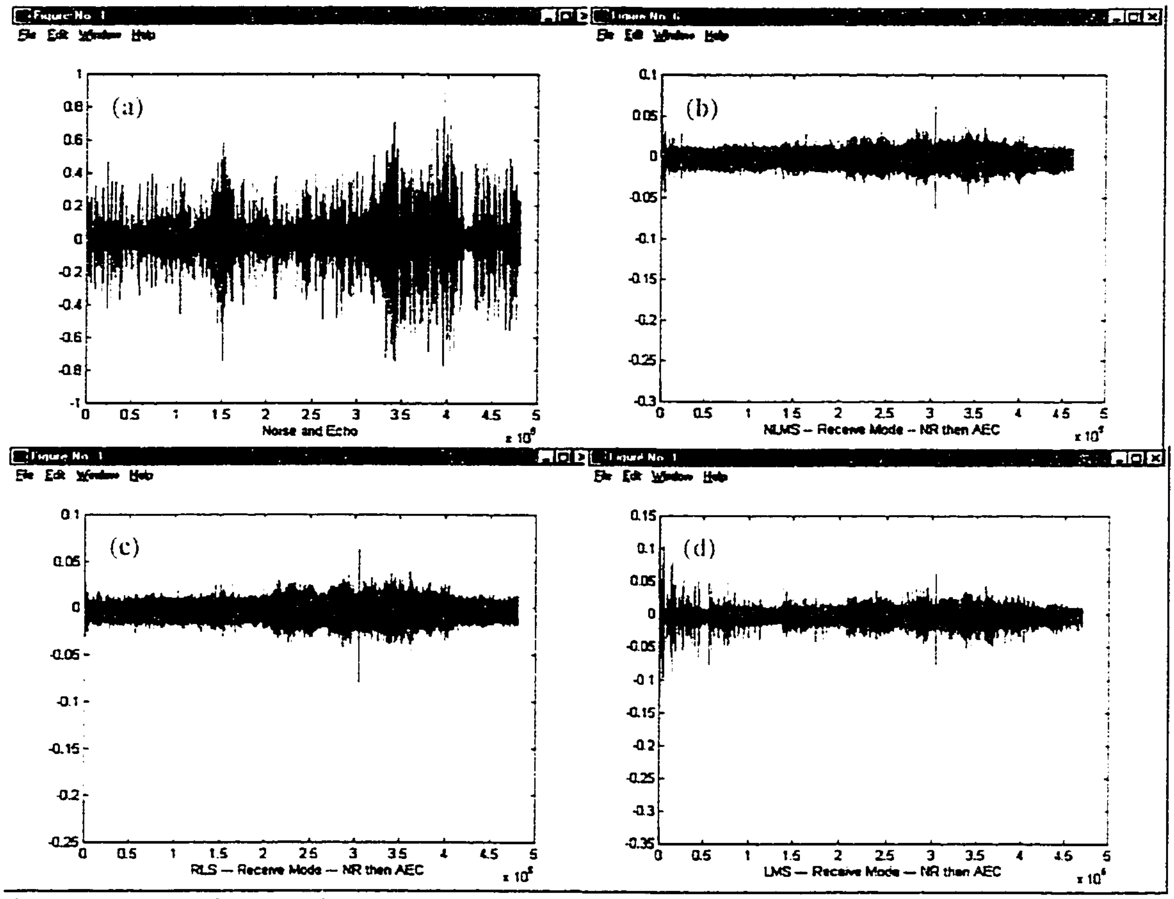

Figure 6.4 Receive Mode

Input datas:

- Primary: noise + echo.
- Reference: noise +echo.

Figure 6.4 (a), (b), (c) and (d) are primary input, NLMS output, RLS output and LMS output respectively. This is more complex computation than the previous mode, because both noise and echo are present at the same time in both primary and reference microphones. A near-end speech signal is not present in this receive mode; therefore, the output of the system should be a silent signal in the ideal situation. From the figure 6.4 (a), it is easy to see that the signal is between [-0.8, 0.8]. The ranges of the output signals of the three algorithms are between [-0.05, 0.05]; they have shown an acceptable results. The LMS slow convergence speed is easily to be seen here.

3. Transmit Mode

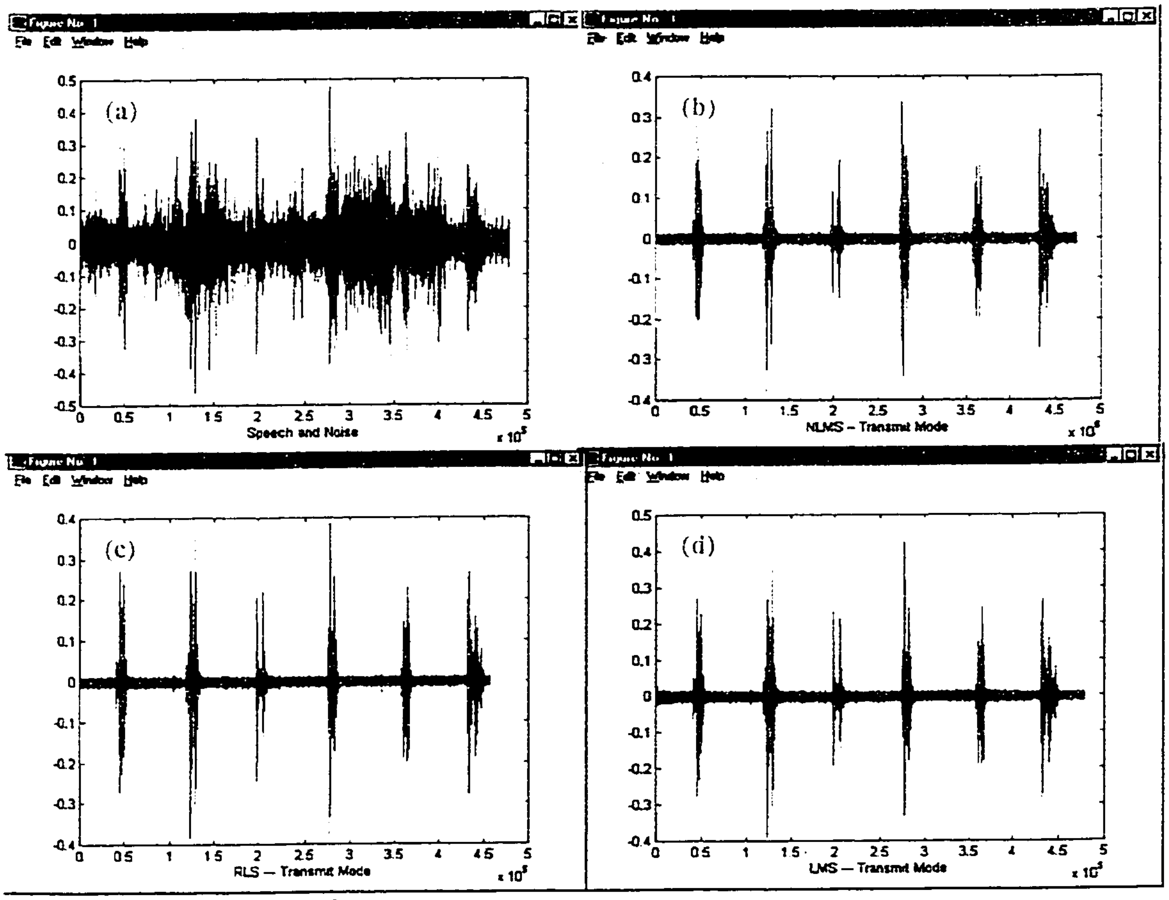

Figure 6.5 Transmit Mode

Input data:

- Primary: noise + speech.
- Reference: noise.

The figure 6.5 (a), (b), (c) and (d) are primary input, NLMS output, RLS output and LMS output respectively. This is an easier mode to deal with, because it contains the primary signal which is corrupted by background noise only. The noise reduction system is here to take advantage of reference noise to produce the close replca of the noise in the primary signal.

From the NLMS, RLS and LMS output figures, the outputs are showing that the system performed a good result in all three algorithms. Especially, LMS is performing better comparing to both receive mode and idle mode; this is because paths of echo are more complex than noise.

4. **Double-Talk Mode**

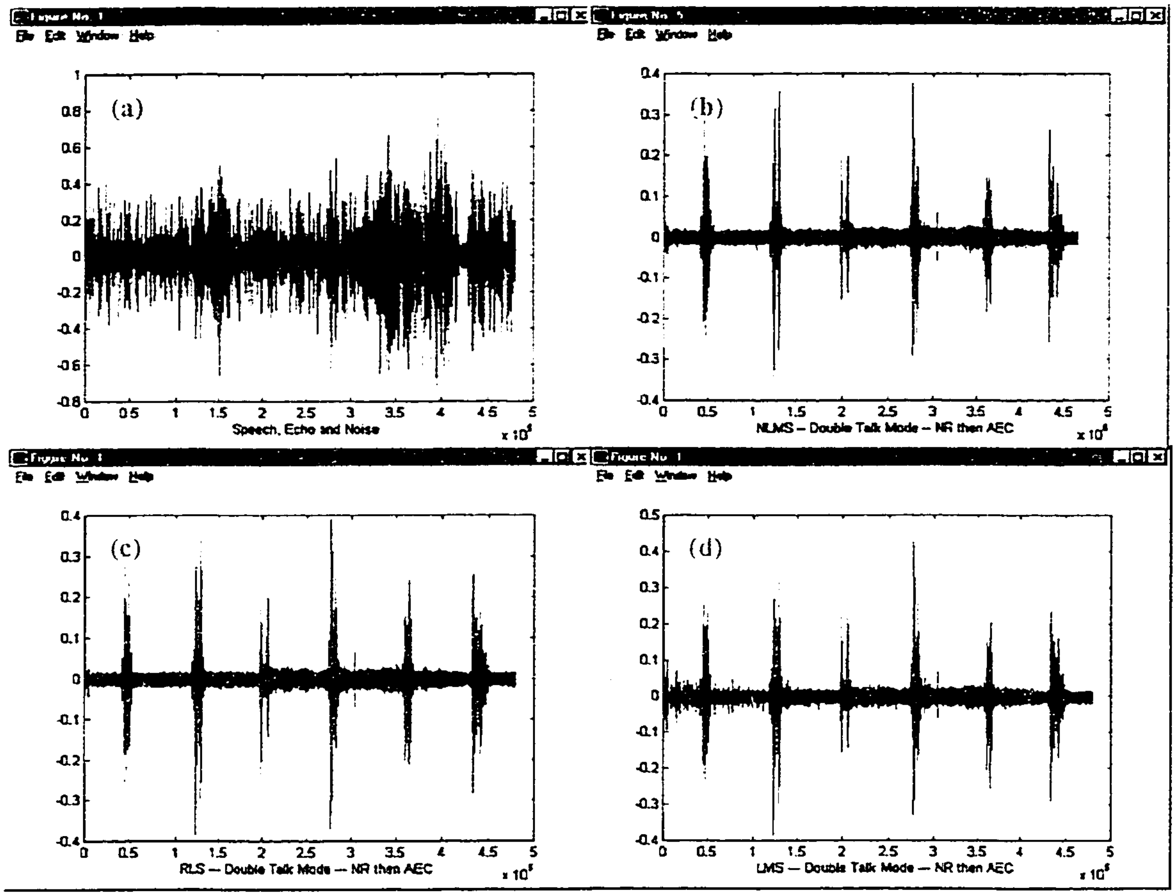

Figure 6.6 Double-Talk Mode

Input Data:

- Primary: noise + echo + speech.
- Reference: noise +echo.

The Figure 6.6 (a), (b), (c) and (d) are primary input, NLMS output, RLS output and LMS output respectively.

Double-talk mode is the most critical state in adaptive interence cancelling system. The complexity of the primary signal and reference signal are always a difficult task to tackle.

A NR filter is used in the first part of the system, and the output of the first filter should be near-end speech and residue of noise and echo. The task of the second filter is cancelling out the residue echo. The final system output should be near-end speech and some residue noise.

The output of the NLMS, LMS and RLS algorithm is quite acceptable compared to the primary corrupted signal. Because of the complexity of the signals, the LMS still has some amount of echo residue.

Signal-to Noise Ratio

The signal to noise ratio (SNR) in acoustic echo cancellation, transmit mode, double-talk mode are shown on the figure 6.7. The input and output signal to noise ratios (SNR) are computed by the two formulas: input $SNR=10\log(S^2/\sigma^2)$, and output $SNR=10\log(S^2/\sigma_o^2)$, where S^2 is the power of pure speech signal, σ^2 is the input noise variance, σ_o^2 is the average of the output square.

On these three systems, the output SNRs are showing good SNR improvement comparing to the input SNRs. To compare the LMS and RLS algorithms, in this figure 6.7, the RLS algorithm reached a better output SNR than the NLMS and LMS algorithms, especially in double-talk mode.

SNR

1. **SNR = 10*log(speech^2/output^2)**
2. **Filter order = 64**

Primary sensor SNR in DT = -18.74 dB

Corrupted by Noise SNR = -6.10 dB

Corrupted by Echo SNR = -8.01 dB

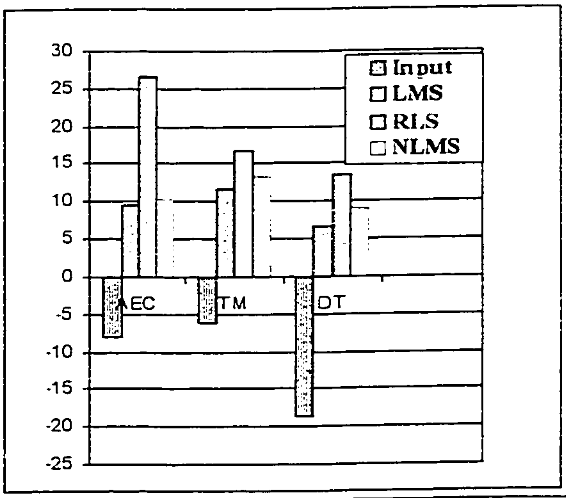

Figure 6.7 Signal to Noise Ratio (SNR)

LMS Step Size

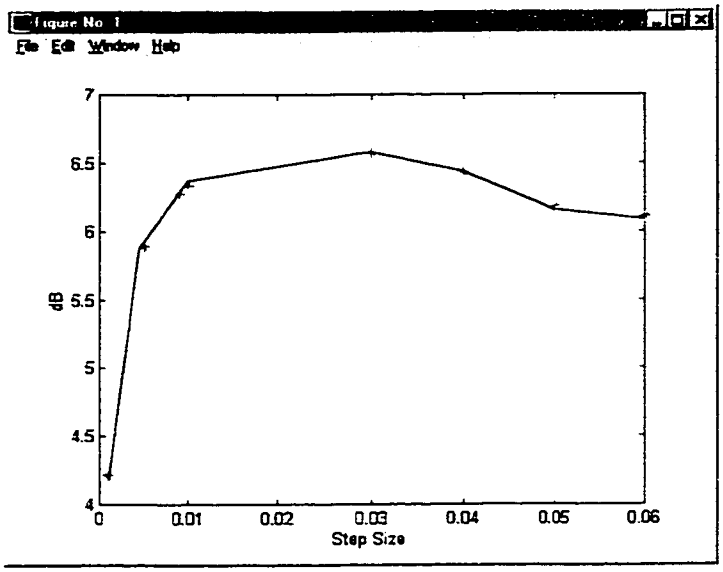

Figure 6.8 SNR v.s. Step Size in LMS

Choice of step size is one of the factors which affects the output SNR of the LMS algorithm.

This figures are showing step size μ vs. output SNR. It is easy to see that when μ = 0.03 the output SNR has the best number among many different step size μ.

The step size,μ , determines the speed of convergence and the stability of the algorithm. The value of is chosen large at the beginning and then is progressively reduced to a smaller size to iterate closer to the optimum value. [3]

Stability of the LMS Algorithm

- It can be shown that starting with an arbitrary initial weight vector, LMS algorithm will converge in the mean and will remain stable as long as the step size (μ) is in the range

  $0 < \mu < 1/(N*power\_of\_reference)$

- which is an easy boundary to calculate.

- Within that margin, the larger μ, the faster the convergence but the less the stability around the minimum value. On the other hand, the smaller the value of u, the slower the convergence, but the output will be more stable around the optimum value.

6.3.3 Combination of Four States in a System

This is a simulation of IDLE, Receive, Transmit and Double Modes in a signal system. The duration is 6.5 seconds; the figure below is the primary corrupted signal.

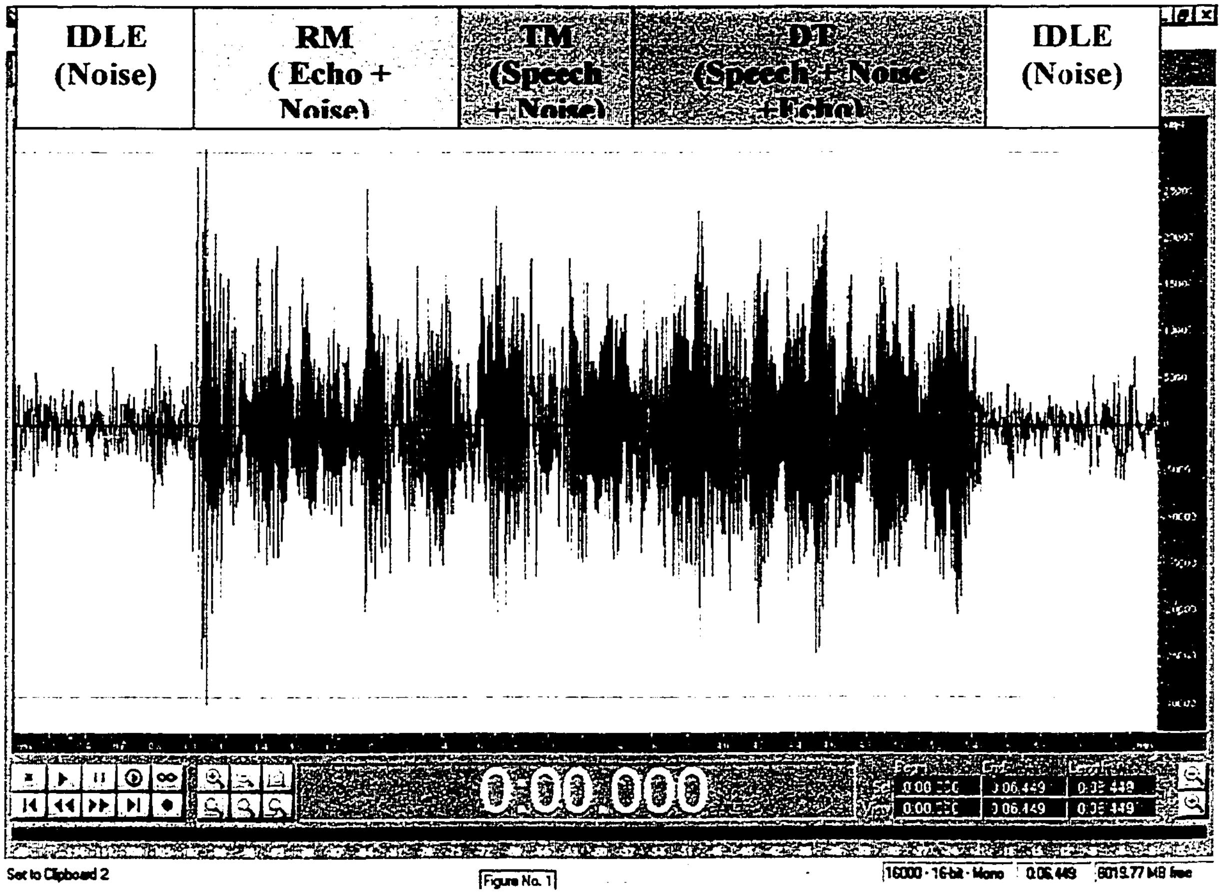

Figure 6.9 Primary input in second set test data

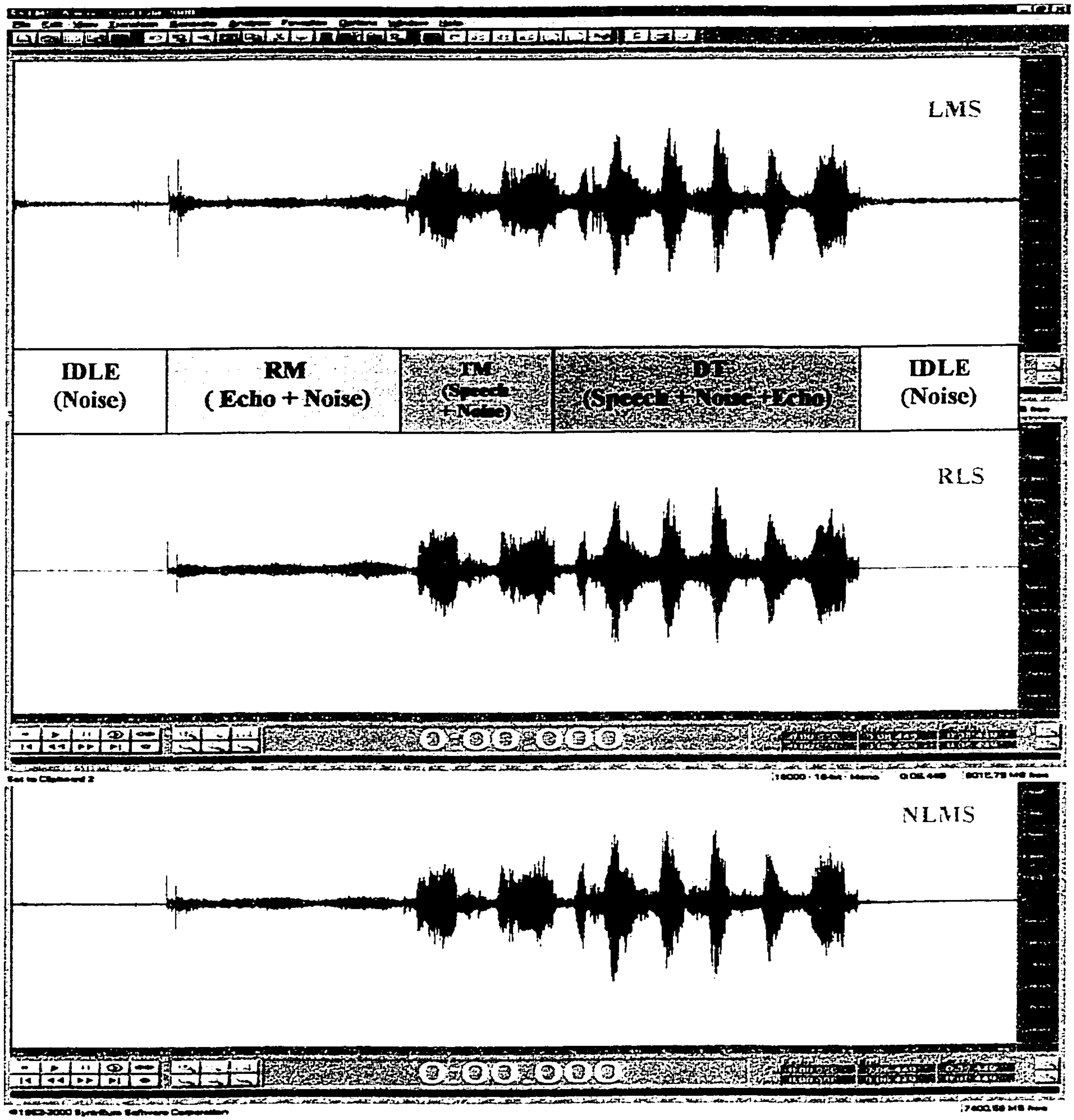

Figure 6.10 Output in combination of four modes

Again, the above figure is showing the comparison of three algorithms, LMS, RLS and NLMS. From convergence speed and output SNR point of view, the RLS is characterized by better stability and higher output SNR as well as faster convergence speed than the LMS algorithm.

But the disadvantage of the RLS is the computation complexity, the program running time is much longer that of the LMS and NLMS algorithms.

Filter Order RLS v.s. LMS

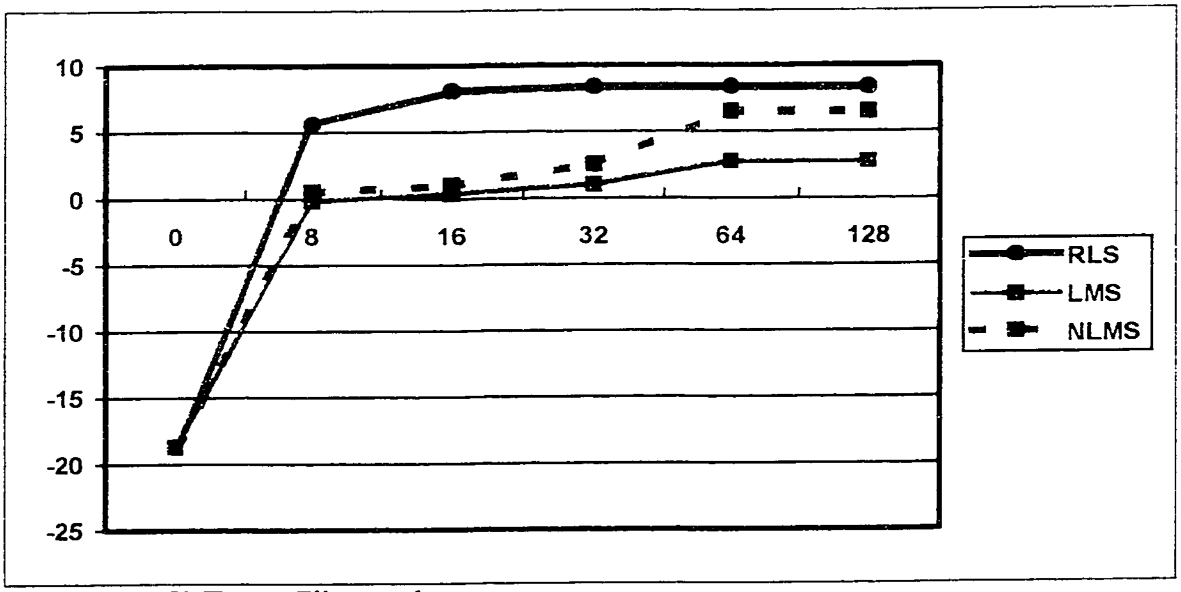

Figure 6.11 SNR v.s. Filter order

The filter order vs. output SNR is showing on the above figure; 6.11. RLS LMS and NLMS algorithms output SNRs has been tried in filters order in 8, 16, 32, 64, 128. For the RLS algorithm, when the order is 32, the SNR has reached the optimum. For the LMS and NLMS algorithms, because of the simple algorithm, the SNR reached the optimum at the order 64.

Convergence Speed

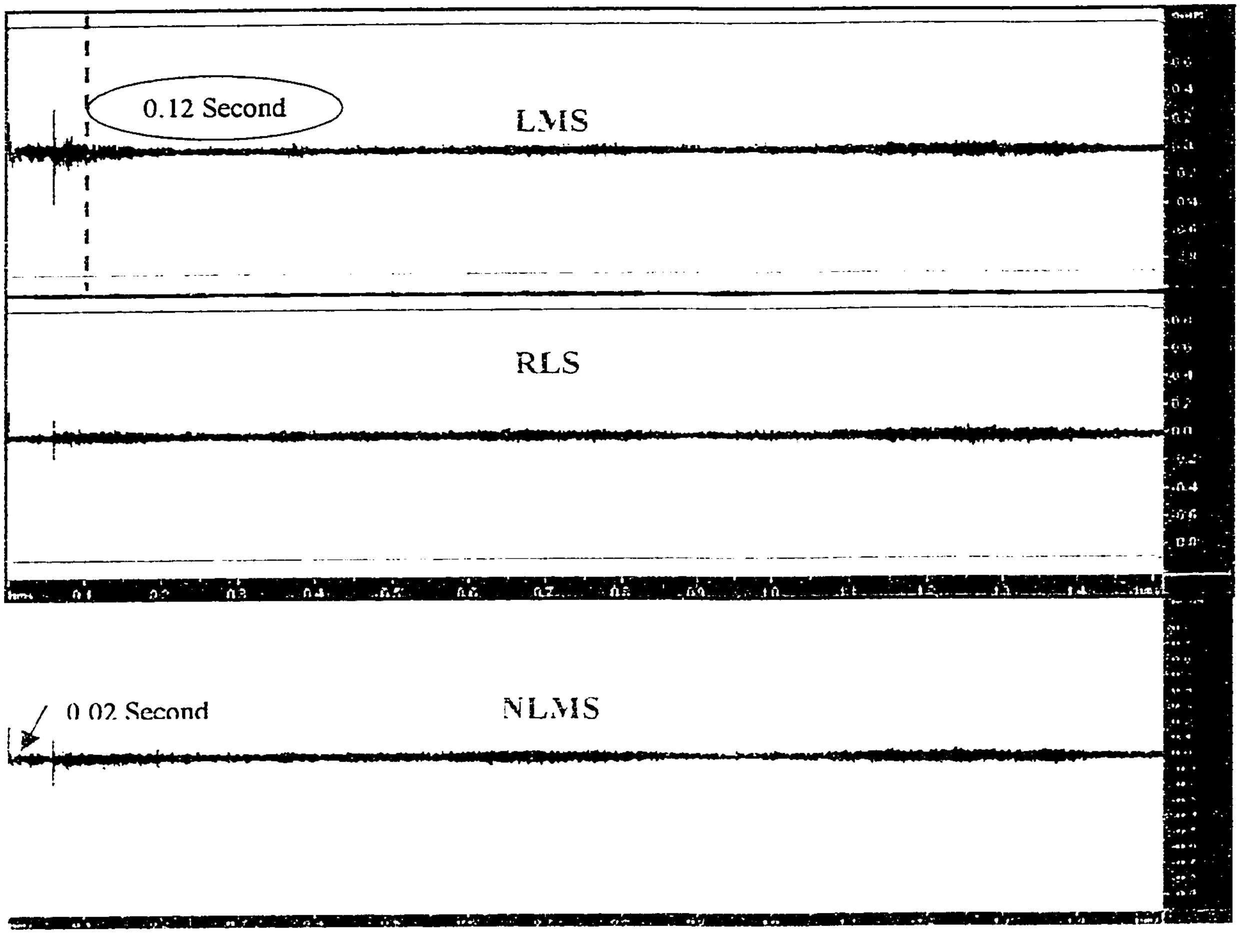

Figure 6.12 Convergence Speed

The above figure 6.12 is comparing convergence speed in the LMS, RLS and NLMS algorithms. The RLS is clearly many times faster than LMS.

The summary of RLS v.s. LMS

LMS v.s. RLS v.s. NLMS			
Filter order –> 64 Corrupted signal -18.73 dB	SNR	Convergence Rate	Computation time in C/C++
LMS	2.74 dB	Around 1900 samples	10 seconds
RLS	8.73 dB	After a first few Samples	20 minutes
NLMS	6.49 dB	Around 720 Samples	15 seconds

Table 6.3 Summary of LMS, NLMS and RLS in SNR, Convergence Speed and Computation time

This table 6.3 is a brief summary of the LMS, RLS and NLMS algorithms in SNR, Convergence Rate, and Program Running time. This shows that the weakness of the LMS lies in poor performance and slow convergence speed by comparing the other two. RLS has a good performance result and fast convergence speed, but the complexity of the algorithm makes the computation time become a weakness. The NLMS is clearly a major improvement in all three aspects.

Chapter 7

7 Conclusions

7.1 Conclusions

The introduction of mobile telephones in cars made hands-free units a safety requirement. Due to the low signal-to-noise ration at the microphone input, inclusion of a noise reduction system became desirable. In hands-free telephony, echo occurs because an open-air acoustic path exits between loudspeaker and microphone. This is extremely complex during periods of double talk. Therefore, in order to be applicable in the considered hands-free telephony environment, the system are required to perform a significant acoustic echo cancellation and noise reduction for a wide range of signal-to-noise ratio conditions. Furthermore, a high near-end speech quality and a natural sounding residual background noise are of great importance.

Instead of using traditional spectral subtraction, adaptive interference cancelling filters are used. If the input SNR is low the adaptive filter will show a better output SNR improvement. We have shown the acoustic echo cancellation, noise reduction separately and a system of combining AEC and NR in two critical states, Receive and Double-Talk. In this thesis we have shown how the tasks of acoustic echo cancellation and noise reduction can be combined.

Adaptive Finite impulse response (FIR) filters for the cancellation are used in the simulation system. Three kinds of algorithms, least-mean-square (LMS) error, recursive-least-square (RLS) error and normalized least-mean-square (NLMS), have been used to adapt the tap weights (coefficients). Comparison of LMS, RLS and NLMS in convergence speed, output SNR, and computation time are carried out to determine the advantages and disadvantages of both algorithms.

The results have shown good improvement in output SNR with the adaptive FIR filter. The RLS filter is a recursive implementation of the Wiener filter. The main advantage of the LMS is the relative simplicity of the algorithm. However, the LMS has an uneven and slow rate of convergence. The rate of convergence of the RLS algorithm is typically faster than that of LMS and NLMS algorithms. RLS output SNR has exhibited good improvement especially in the critical double-talk state, but due to the complexity of the RLS algorithm, the computation time is extremely long compared to the LMS and NLMS algorithms. NLMS is showing a faster convergence rate and better output performance than LMS and also retaining the advantage of LMS, short computation time.

In conclusion, adaptive interference systems have done their jobs on cancelling unwanted signals or noise as much as possible. LMS, RLS and NLMS algorithms are all performing with reasonable output quality in either AEC or NR. However, the RLS algorithm is showing a better result in more complex and more critical two states: receive and double talk.

7.2 Future Work

The future work of our study of adaptive filter in acoustic echo cancellation and noise reduction in LMS, RLS and NLMS algorithms may be focused on two directions: 1. further study on the residues of noise and echo, 2. and maybe an ideal algorithm that has RLS algorithm's output quality and fast convergence rate and the LMS algorithm computation time.

The residue of noise and echo is still present in two major critical states: receive and double-talk. If residue is quite noticeable, it may degrade the quality of hands-free telephony system. Although we have used AEC filter to cancel out the possible echo residue after the first NR filter, the noise residue still remains.

Hardware realization is always a big part of any system design and algorithm implementation. Because the simplicity of LMS algorithm, it has already been used in some companies. RLS computation time and complexity is not easy to be done in real time hardware implementation.

Taking the advantages of RLS such as output quality and convergence speed as well as the advantages of LMS's simplicity and its program running time into consideration, a balance between RLS and LMS could be achieved in the future. In other word, the goal is to find a simpler RLS without sacrificing its advantages, or a improved LMS and NLMS algorithms.

References

[1] Stearns, Samuel D. and Hush, Don R., Digital Signal Analysis, 2nd ed., Prentice-Hall, Englewood Cliffs, N.J., 1990.

[2] Rabiner, Lawrence R. and Gold, Bernard, Theory and Application of Digital Signal Processing, Prentice-Hall, Englewood Cliffs, N.J., 1975.

[3] Vaseghi, Saeed V., Advanced Signal Processing and Digital Noise Reduction, John Wiley & Sons, New York, 1996.

[4] Tohyama, Mikio and Koike, Tsunehiko, Fundamentals of Acoustic Signal Processing, Academic Press, San Diego, California, 1998.

[5] Figueeiras-Vidal, Anibal R., Digital Signal Processing in Telecommunications, Springer, London, 1996.

[6] Hayes, Monson H., Statistical Digital Signal Processing and Modeling, John Wiley & Sons, Toronto, 1996.

[7] Haykin, Simon, Adaptive Filter Theory, 3rd ed., Prentice-Hall, Toronto, 1996.

[8] Boll, Steven F., "Suppression of Acoustic Noise in Speech Using Spectral Subtraction", IEEE Trans. on ASSP., VOL 27, No. 2, pp. 113-120, 1979.

[9] Shozakai, M., Nakamura, S. and Shikano, K., "Robust Speech Recognition in Car Environments", IEEE ICASSP, pp 269-272, 1998.

[10] Sankaran Sundar G. and Beex, A. A. (Louis), "Acoustic Echo and Noise Canceller Improvements For Hand-Free Telephones".

[11] Kuo, S. M., Huang, Y. C. and Pan, Z., "Acoustic Noise and Echo Cancellation Microphone System For Video-Conferencing", IEEE Trans. on Consumer Electronics, Vol. 41, No. 4, pp 1150-1158, Nov. 1995.

[12] Pascal, S. and Benamar, A., "A System for Speech Enhancement in the Context of Hands-Free Radiotelephony with combined Noise Reduction and Acoustic Echo Cancellation", Speech Communication, Vol. 20, pp 203-214, 1996.

[13] Martin, R. and Gustafsson, S., "The Echo Shaping Approach to Acoustic Echo Control", Speech Communication, Vol. 20, pp 181-190, 1996.

VITA AUCTORIS

NAME: Wayne Hui-Chung Chiang

PLACE OF BIRTH: TAIWAN, R.O.C.

DATE OF BIRTH: May, 1969.

EDUCATION: B.A.Sc.
Department of Electrical Engineering
University of Windsor
Windsor, Ontario,
1992 – 1996

M.A.Sc.
Department of Electrical Engineering
University of Windsor
Windsor, Ontario,
1997 – 2000